

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО**

Факультет інформатики та обчислювальної техніки

(назва факультету, інституту)

Кафедра автоматизованих систем обробки інформації і управління

(назва кафедри)

"На правах рукопису"

УДК 004.93(015.7)

«До захисту допущено»

В.о.завідувача кафедри

О.А.Павлов

(підпис)

(ініціали, прізвище)

“ ” 20 19 р.

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ

на здобуття ступеня магістра

за спеціальністю 126 Інформаційні системи та технології

(код та назва спеціальності)

ОПП

Інформаційні управляючі системи та технології

(код та назва спеціалізації)

на тему: Система підтримки процесів діяльності комунального транспортного підприємства з вивезення відходів

Виконав: студент

VI курсу групи ІС-81мп

(шифр групи)

Любченко Гліб Олександрович

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник

доц., к.т.н., Сперкач М.О.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант

проф., д.т.н. Томашевський В.М.

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент

доц.каф. АУТС, к.т.н., Писаренко А.В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент

(підпис)

Київ – 2019

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації та управління
(повна назва)

Рівень вищої освіти другий (магістерський) за освітньо-професійною програмою

Спеціальність 126 Інформаційні системи та технології
(код і назва)

ОПП Інформаційні управляючі системи та технології
(код і назва)

ЗАТВЕРДЖУЮ

В.о.завідувача кафедри

О.А.Павлов
(підпис) (ініціали, прізвище)

«__» _____ 2019 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Любченку Глібу Олександровичу

(прізвище, ім'я, по батькові)

1. Тема дисертації Система підтримки процесів діяльності комунального транспортного з вивезення відходів

науковий керівник дисертації Сперкач Майя Олегівна, доц., ктн.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “ 28 ” 10 20 19 р. № 3770-с

2. Строк подання студентом дисертації “ 2 ” 12 20 19 р.

3. Об'єкт дослідження оптимізація задачі складання розкладу перевезень шляхом рівномірного розподілу праці

4. Перелік завдань, які потрібно розробити 1) виконати огляд існуючих методів та засобів складання розкладу перевезень для транспортних компаній;
2) формалізувати задачу складання розкладу перевезень в умовах недовизначеності показників; 3) розробити програмне забезпечення на основі зазначених вище методів;
4) виконати аналіз отриманих результатів.

5. Орієнтовний перелік ілюстративного матеріалу 1) схема структурна варіантів використання; 3) ER-діаграма бази даних, результати експериментів; 4) діаграма пакетів; 5) діаграма послідовностей алгоритму складання розкладу перевезень; 6) діаграма послідовностей алгоритму кластеризації; 7) діаграма послідовностей алгоритму розв'язання задачі комівояжера; 8) архітектура системи.

6. Орієнтовний перелік публікацій стаття у міжвідомчому науково-технічному збірнику «Адаптивні системи автоматичного управління» №2/33, тези доповіді у міжнародному науковому симпозіумі «Інтелектуальні рішення», тези доповіді у всеукраїнській науково-практичній конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2019).

7. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

8. Дата видачі завдання “ 2 ” вересня 20 19 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Систематизація результатів огляду літератури	05.09	
2	Порівняльний аналіз існуючих методів розв'язання задачі	12.09	
3	Постановка та формалізація математичної моделі задачі	17.09	
4	Модифікація існуючих методів розв'язання задачі	25.09	
5	Розробка інформаційного та програмного забезпечення	02.10	
7	Проведення експериментальних досліджень розроблених алгоритмів	18.10	
8	Оформлення документації	01.11	
9	Подання роботи на попередній захист	20.11	
10	Подання роботи на основний захист	02.12	

Студент

(підпис)

Г.О. Любченко

(ініціали, прізвище)

Науковий керівник

(підпис)

М.О. Сперкач

(ініціали, прізвище)

РЕФЕРАТ

Магістерська дисертація: 105 с., 12 рис., 35 табл., 66 джерел, 1 додаток.

Актуальність. У 2017 році уряд України розробив Національну стратегію з управління відходами до 2030 року. Стратегія була створена для покращення сфери поводження з відходами в Україні до європейських стандартів, що є однією з умов вступу нашої держави в Євросоюз. За дослідженнями Міністерства регіонального розвитку, будівництва та житлово-комунального господарства майже чверть населення країни досі не охоплена послугою з вивезення сміття. Сфера послуг поводження з побутовими відходами наразі дуже проблемна. Це пов'язано зі збільшенням обсягу утворення побутових відходів, в Україні за рік одна людина створює 250-300 кілограмів на рік. Саме тому варто приділити увагу проблемам з управлінням відходами, доки Україна буквально не потонула у смітті.

Через процеси глобалізації великі міста розростаються все з більшою швидкістю, що в свою чергу означає стрімке збільшення точок збору сміття, які повинні регулярно вивозитися. Таке збільшення точок вивезення сміття призводить до розростання розмірності транспортної задачі, яку компанії, які займаються вивезення сміття повинні вирішувати кожен день. Задача об'їзду всіх точок по колу та повернення в початковий пункт відноситься до класу NP-повних, що означає що знаходження оптимального рішення будь-яким іншим способом окрім як методом повного перебору ще не знайдено. Саме тому, кожна оптимізація, яка покращує цей процес хоча б на якийсь відсоток є актуальною та корисною. В даній роботі розглянуто спосіб оптимізації подібної транспортної задачі за часовим параметром, шляхом рівномірного розподілу роботи на етапі планування.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалась на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» в рамках теми «Ефективні методи розв'язання задач теорії розкладів. Державний реєстраційний номер» (№ ДР 0117U000919).

Мета дослідження – підвищення ефективності функціонування комунального транспортного підприємства за рахунок побудови оптимальних або близьких до оптимальних розкладів перевезень.

Для досягнення мети необхідно виконати наступні **завдання**:

- виконати огляд існуючих методів та засобів складання розкладу перевезень для транспортних компаній;
- формалізувати задачу складання розкладу перевезень в умовах недовизначеності показників;
- розробити програмне забезпечення для системи складання розкладу перевезень на основі зазначених вище методів;
- виконати аналіз отриманих результатів.

Об'єкт дослідження – оптимізація задачі складання розкладу перевезень шляхом рівномірного розподілу праці.

Предмет дослідження – система підтримки процесів діяльності комунального транспортного підприємства з вивезення відходів.

Методами дослідження є розв'язання транспортної задачі та задачі кластеризації.

Наукова новизна одержаних результатів полягає у розробці нового методу рівномірного розподілу всього об'єму робіт між виконавцями, при цьому не погіршуючи показників відстані.

Публікації. Матеріали роботи опубліковані в міжвідомчому науково-технічному збірнику «Адаптивні системи автоматичного управління» №2/33, у тезах міжнародного наукового симпозіуму «Інтелектуальні рішення» та у всеукраїнській науково-практичній конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2019).

ЛОГІСТИКА, ЗАДАЧА КОМІВОЯЖЕРА, НЕДОВИЗНАЧЕНІ ПОКАЗНИКИ, ТРАНСПОРТНА ЗАДАЧА, АЛГОРИТМИ СКЛАДАННЯ РОЗКЛАДІВ ПЕРЕВЕЗЕНЬ, ЗАДАЧА КЛАСТЕРИЗАЦІЇ

ABSTRACT

Master's Thesis: 105 pp., 12 images, 35 tables, 66 sources, 1 supplement. **Topicality.** In 2017, the Government of Ukraine has developed a National Waste Management Strategy by 2030. The strategy was created to improve the sphere of waste management in Ukraine to European standards, which is one of the conditions for our country's accession to the European Union. According to the Ministry of Regional Development, Construction and Housing, almost a quarter of the country's population are still not covered by the garbage collection service. The area of household waste management is currently very problematic. It is caused by the increase in the volume of household waste generation, in Ukraine, one person creates 250-300 kilograms per year. Therefore, it is worth paying attention to waste management issues until Ukraine literally sank.

Through globalization, large cities are expanding at an increasing rate, which in turn means a rapid increase in garbage collection points that need to be removed regularly. This increase in the number of garbage collection points leads to an increase in the transport problem, which garbage collection companies have to deal with every day. The task of bypassing all points in a circle and returning to the starting point belongs to the class of NP-complete, which means that finding the optimal solution by any means other than a complete search has not been found yet. Therefore, any optimization that enhances this process by at least one percent is relevant and useful. This paper discusses a way to optimize a similar transport task by time parameter, by evenly distributing the work during the planning stage. **Relationship of work with scientific programs,** plans, themes. The work was carried out at the Department of Automated Systems for Information Processing and Management of the National Technical University of Ukraine "Kyiv Polytechnic Institute. Igor Sikorsky" within the framework of the theme "Effective methods for solving the problems of the theory of schedules", state registration number 0117U000919.

The purpose of the study is to increase the efficiency of the municipal transport enterprise by building optimal or close to optimal schedules.

To achieve the goal, you must accomplish the following **tasks**:

- analysis of existing transport planning methods and tools for transport companies;
- to formalize the task of transport planning in the conditions of uncertainty of indicators;

- development of software for a transportation schedule system based on the above methods;
- analyze the results.

The object of research is to optimize the task of compiling a transportation schedule by means of a uniform distribution of labor.

The subject of the research is the system of support of processes of activity of the enterprise of city transport on waste transportation.

The research methods used in the work are based on solving transportation theory and cluster analysis.

The scientific novelty of the obtained results is to develop a new method of evenly distributing the entire volume of work between the performers, without compromising the distance.

Publications. The materials of the work were published at the scientific and technical conference "Adaptive automatic control systems" №2 / 33, in the abstracts of the international scientific symposium "Intelligent solutions" and in the all-Ukrainian scientific and practical conference of young scientists and students "Information systems and control technologies" (ISTU- 2019).

LOGISTICS, TRAVELING SALESMAN PROBLEM, UNDEFINED INDICATORS, TRANSPORTATION THEORY, TRANSPORTATION SCHEDULE PLANNING ALGORITHMS, THE CLUSTERING PROBLEM

ЗМІСТ

1 ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ СИСТЕМИ ПІДТРИМКИ ПРОЦЕСІВ ДІЯЛЬНОСТІ СПЕЦІАЛІЗОВАНОЇ ТРАНСПОРТНОЇ КОМПАНІЇ	9
1.1 Опис бізнес-процесів	9
1.1.1 Опис процесу діяльності	10
1.1.2 Актори і функції	12
1.1.3 Структура бізнес-процесів	15
1.2 Опис постановки задачі	20
1.3 Огляд аналогів системи, що розробляється	21
1.4 Рішення з інформаційного забезпечення	23
Висновок до розділу	29
2 МОДЕЛІ ТА МЕТОДИ ПІДТРИМКИ ПРОЦЕСІВ ДІЯЛЬНОСТІ СПЕЦІАЛІЗОВАНОЇ ТРАНСПОРТНОЇ КОМПАНІЇ	31
2.1 Змістовна постановка задачі	31
2.2 Огляд методів розв'язання	32
2.2.1 Методи розв'язання задачі комівояжера	32
2.2.3 Методи розв'язання задачі кластеризації	35
2.2.3 Методи розв'язання задачі планування розкладу перевезень	35
2.3 Огляд існуючих рішень	39
2.4 Розв'язання транспортної задачі з рівномірним розподілом роботи	40
2.4.1 Постановка задачі	40
2.4.2 Дослідження властивостей задачі	42
2.4.3 Алгоритм розв'язання задачі	43
2.5 Експериментальні результати досліджень	48
Висновок до розділу	53
3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	54

	8
3.1 Засоби розробки	54
3.2 Вимоги до технічної частини	55
3.3 Вимоги до програмного продукту	56
3.4 Архітектура програмного забезпечення	57
3.4.1 Клієнтська частина	58
3.4.2 Серверна частина	62
3.5 Інструкція користувача	63
Висновок до розділу	67
4 РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ	68
4.1 Опис ідеї проекту	68
4.2 Опис конкурентів	69
4.3 Технологічний аудит ідеї проекту	72
4.4 Аналіз ринкових можливостей запуску стартап-проекту	73
4.5 Розроблення ринкової стратегії проекту	81
4.6 Розроблення маркетингової програми стартап-проекту	84
Висновок до розділу	88
ВИСНОВКИ	89
ПЕРЕЛІК ПОСИЛАНЬ	90
ДОДАТКИ	96
ДОДАТОК А	96

1 ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ СИСТЕМИ ПІДТРИМКИ ПРОЦЕСІВ ДІЯЛЬНОСТІ СПЕЦІАЛІЗОВАНОЇ ТРАНСПОРТНОЇ КОМПАНІЇ

1.1 Опис бізнес-процесів

Наразі процеси пов'язані з вивезенням сміття є одними з найважливіших для підтримки комфортного рівня життя та стану соціального добробуту населення, адже якщо цей процес зупиниться хоча б на один тиждень в місті почнеться санітарно-епідеміологічна катастрофа.

Саме тому ми не можемо допустити припинення процесу вивезення сміття хоча б на день. Зі збільшенням населення великих міст, та відповідно кількості об'єктів, з яких необхідно відвантажувати сміття, задача організації всього цього процесу починає значно ускладнюватися. Людина більше не в змозі будувати оптимальні маршрути для транспортних засобів, які цим займаються. Вже при розмірності задачі в 10 пунктів збору ми маємо 181440 унікальних варіантів маршруту, що вже казати про те, коли пунктів десятки, а може навіть сотні тисяч, які одночасно обслуговують тисячі машин.

Людина здатна розв'язати ці проблеми тільки з використанням інформаційних технологій. Однак, вже розроблені рішення мають певні недоліки, такі як:

- неоптимальність побудови маршруту;
- відсутній алгоритм, який міг би автоматично займатися розв'язанням питання рівномірного розподілу роботи між водіями (для того, щоб всі закінчили роботу приблизно одночасно);
- забагато ризиків пов'язаних з людським фактором;
- проблема невизначеності показників кожного автомобіля (передбачити скільки в нього поміститься контейнерів можна лише з певною похибкою).

Важливість цього дослідження полягає в тому, що воно дозволить зменшити матеріальні витрати шляхом скорочення відстані, а відповідно і витраченого палива, також уникнути марнування часу. Це у свою чергу збільшить пропускну спроможність подібних компаній. Внаслідок чого, більша кількість населення буде забезпечена сервісом та зростатимуть прибутки. Також, у випадку повної автоматизації процесу, ризики пов'язані з людським фактором, наприклад певні об'єкти можуть бути не опрацьовані вчасно, будуть зменшені. Значно спроститься

процес додавання нових об'єктів тому, що більше не потрібно буде виконувати статичне планування маршрутів.

1.1.1 Опис процесу діяльності

На даний момент більшість компаній, залучених в дану предметну галузь, працюють наступним чином:

- компанії укладають угоди та беруть на баланс об'єкти обслуговування (точки вивезення сміття);
- складається розклад вивезення;
- відбувається умовний поділ на сектори;
- в кожному секторі вирішується транспортна задача;
- сектори закріплюються за автомобілем, який буде їх опрацьовувати;
- автомобіль починає здійснювати рейси в межах свого сектору доки всі контейнери не будуть опрацьовані.

На рисунку 1.1 цей процес зображено у вигляді структурної схеми.



Рисунок 1.1 – Схема структурна роботи системи до автоматизації

Основними очевидними недоліками даного методу є:

- відсутність механізмів, які забезпечують одночасне завершення процесу обробки для водіїв. В результаті виникнення ситуації, якщо один водій впорався швидше ніж інший, необхідне втручання оператора для перерозподілу роботи між ними;
- якщо в машині після завершення обробки свого сектору ще залишилося вільне місце то знову стає необхідним людське втручання для перерозподілу маршрутів;
- оскільки перерозподіли відбуваються з людським втручанням, то це потенційно може стати джерелом помилок та не раціональних рішень, які у свою чергу призводять до збільшення відстаней і часу обробки;
- система архітектурно не готова бути гнучкою, наприклад важко додати функціонал, який би дозволяв опрацьовувати контейнери не по статичному розкладу, а по мірі надходження заявок.

У випадку застосування методів, що пропонуються в даному дослідженні, система після її повної автоматизації буде працювати наступним чином:

- автоматично спрацьовує планувальник задач, який серед всіх контейнерів, на збір яких поступили заявки, автоматично будує сектори, їх кількість дорівнює кількості машин, які вийшли на зміну;
- після цього будується маршрут і водії починають свою зміну;
- після того як будь-хто з водіїв закінчує свою поїздку і відправляється на базу відвантажуватися, йде перевірка необхідності виконати перерозподіл секторів, у випадку якщо хтось з водіїв починає відставати;
- якщо система виявляє, що розподіл необхідно провести, то вона автоматично зменшує кількість об'єктів у водіїв, які відстають, і додає їх тим, хто впорався швидше.
- система фіксує всю статистику виконання робіт, що дозволяє проводити аналізи та вимірювати рівень продуктивності різних бригад.

На рисунку 1.2 цей процес зображено у вигляді структурної схеми.

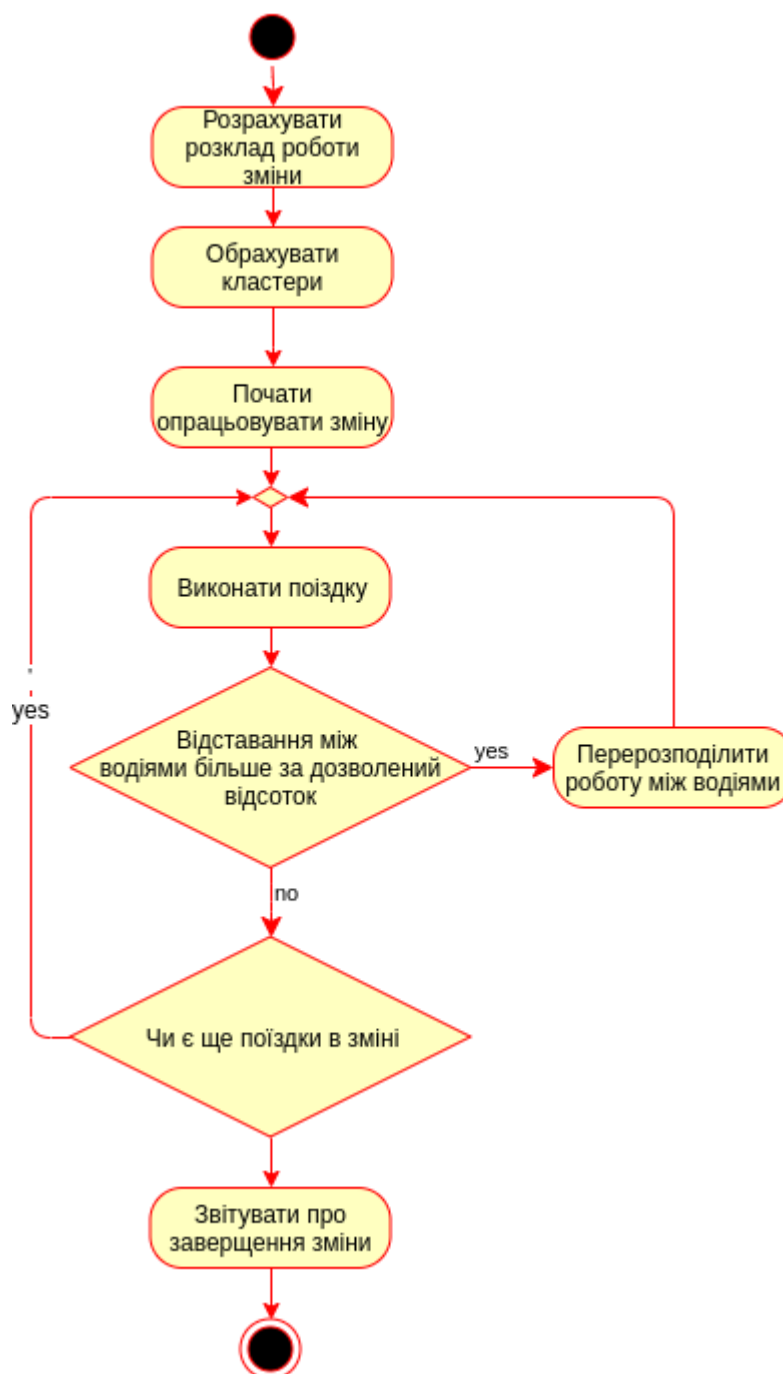


Рисунок 1.2 – Схема структурна роботи системи після автоматизації

Отже, як ми можемо побачити, автоматизація в даній сфері здатна усунути багато недоліків, які має поточна реалізація.

1.1.2 Актори і функції

Опишемо функціональну модель системи та її способи взаємодії з користувачами за допомогою діаграми варіантів використання, яка представлена в додатку А на плакаті 1.

Основні діючі актори в системі є:

- адміністратор;

- водій;
- локальний адміністратор;
- керівництво компанії.

Визначимо перелік дій та функцій, які кожен з представлених акторів може виконувати:

- водій:
 - 1) отримати маршрут;
 - 2) опрацювати маршрут;
 - 3) звітувати про кожен етап виконання;
- локальний адміністратор:
 - 1) подавати заявки на вивезення сміття;
 - 2) контролювати стан виконання роботи;
- керівництво компанії:
 - 1) переглядати статистику та обсяг виконаних робіт;
 - 2) аналізувати оптимальність бізнес-процесів;
 - 3) вести контроль фінансової звітності;
- адміністратор системи:
 - 1) керування обліковими записами всіх інших користувачів;
 - 2) керування зміною (які водії та автомобілі будуть брати участь в конкретній робочій зміні);
 - 3) коригування роботи системи у форс-мажорних випадках;

Надамо короткий опис кожному з варіантів використання.

Отримати маршрут – водій заходить у свій особистий кабінет в системі, де він може отримати автоматично сформований для нього план робіт на зміну, який містить перелік об'єктів, які водій має опрацювати, та автоматично побудований маршрут між ними. Після цього він повинен кожен раз обирати наступну ціль свого слідування.

Опрацювати маршрут – після отримання маршруту водій приступає до його виконання, він починає переміщуватися від одного об'єкту до іншого, і так поки не будуть опрацьовані всі об'єкти, які були на нього виділені. Один із можливих

варіантів відстеження водія – це моніторинг його GPS координат, або мануальне звітування по прибуттю на кожний новий об'єкт обслуговування.

Звітувати про кожен етап виконання – по завершенню кожного етапу водій має викликати відповідний функціонал для звітування та отримання наступної цілі, у випадку, коли водій бачить, що машина повна, він також звітує, що відправляється на точку відвантаження, що дає системі змогу в цей момент перерахувати маршрути для інших водіїв.

Подавати заявки на вивезення сміття – людина, обрана відповідальною за стан заповненості контейнерів на своїй локальній території, має звітувати в своєму особистому кабінеті тоді, коли бачить, що контейнери наповнились та потребують вивезення.

Контролювати стан виконання роботи – локальний адміністратор – та роль, яка визначає рівень задоволеності кінцевих споживачів послуги (мешканців відповідних будинків, які уклали контракт з компанією – користувачем даної системи). Саме тому ця роль приймає рішення стосовно того продовжувати чи скасовувати контракти на подальшу співпрацю.

Переглядати статистику та обсяг виконаних робіт – керівництво компанії має потребу в формуванні звітності для різних видів контролю підприємства, саме тому їм необхідно мати змогу переглядати статистику та звітність за різними критеріями.

Аналізувати оптимальність бізнес-процесів – завдяки можливості переглядати звіти, сформовані за результатами попередніх виконань, керівництво компанії має змогу керувати різними параметрами оптимізації процесу автоматизації, а саме: зміщувати пріоритети оптимізації в сторону найменшого часу обробки, найменшої відстані або найменшої кількості авто, залучених до обробки зміни.

Вести контроль фінансової звітності – завдяки підсистемі формування звітності керівництво компанії має змогу контролювати витрати на паливо, та справедливо оцінювати рівень заробітної плати для працівників в залежності від обсягу виконаної роботи.

Керування обліковими записами всіх інших користувачів – необхідно мати змогу керувати системою, а саме обліковими записами всіх інших користувачів в

системі, наприклад видавати акаунти новим працівникам, або видаляти їх для тих, хто більше з компанією не співпрацює.

Керування зміною – адміністратор має змогу в ручному режимі керувати складом зміни у випадку, якщо хтось із водіїв або інший персонал за певних обставин не в змозі взяти участь у поточній зміні, або у випадку виходу з ладу транспортних засобів.

Коригування роботи системи у форс-мажорних випадках – на 100% розраховувати на автоматизу в будь-якому випадку не можна, саме тому необхідно мати механізм, який дозволяє людині вклинюватися в цей процес і вносити необхідні зміни.

1.1.3 Структура бізнес-процесів

Покращення якості отримання послуг та фінансових показників компанії виконуються за допомогою проведення реінжинірингу бізнес-процесів компанії. Що в свою чергу спонукає до створення АС, яка автоматизує бізнес-процеси та поліпшує якість взаємодії всіх учасників процесу.

Опис бізнес-процесів

На вхід подається замовлення клієнта, яке може надходити з договору, додатків до нього та офіційних листів, за допомогою яких буде виконуватись отримання послуги. Після цього з клієнтом укладається контракт. Далі після оплати послуги об'єкти клієнта починають опрацьовуватися транспортною компанією.

На виході отримуємо звітність та виконану послугу. Роботи по отриманню послуги виконують декілька учасників процесу, а саме: менеджер, оператор, водій. Всі процеси проходять з дотриманням Законів України та рішень КМДА.

Всі етапи виконання фіксуються в середині АС, та можуть бути використані для моніторингу поточного стану системи.

Описані бізнес-процеси зображено на рисунку 1.3 за допомогою BPMN діаграми.

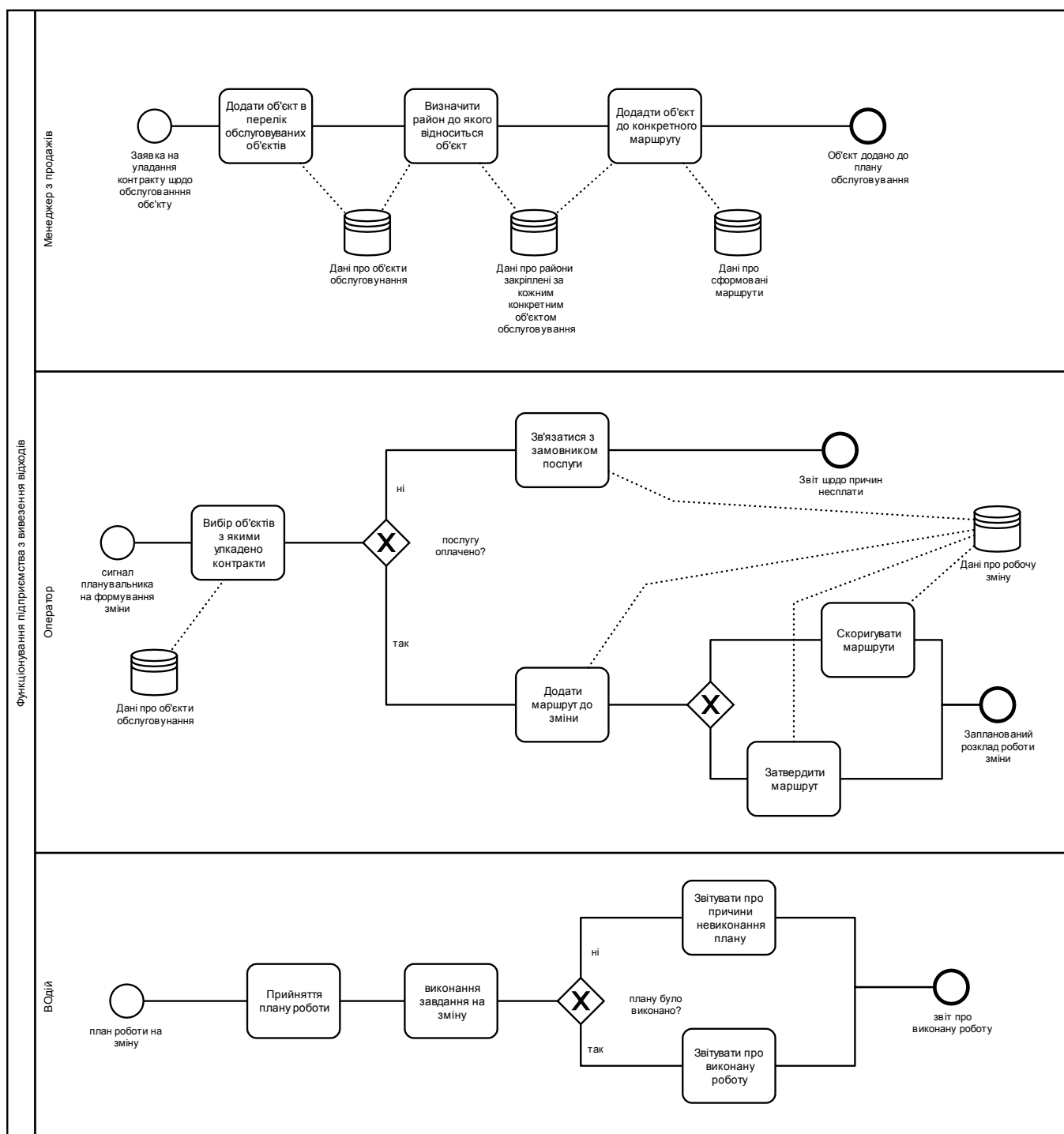


Рисунок 1.3 – Представлення бізнес-процесів укладення контракту, формування робочої зміни та виконання зміни за допомогою нотації BPMN

До автоматизації

Для початку роботи клієнт укладає угоду про послуги з компанією. Менеджер вносить всі відомості з договору та його додатків до бази. На наступному етапі ці відомості надаються диспетчеру, який розносить об'єкти дислокації за маршрутами. Раз на тиждень диспетчер друкує в паперовому вигляді маршрутне завдання на тиждень у вигляді маршрутів для водіїв та розподіляє їх між водіями. На додачу до

маршрутів з завданням на тиждень водій отримує зошит, в якому має записувати всі відомості згідно з фактичним обсягом наданих послуг або причин з яких виконаний обсяг не відповідає запланованому.

Окремим процесом у всій цій діяльності є формування маршрутів диспетчером. Зараз диспетчери вручну розподіляють всі об'єкти на сектори та вносять їх в логістичну систему, яка будує статичні маршрути, а після цього закріплюють ці маршрути за водіями. Виконувати цю роботу вручну дуже важко і неефективно. Це робить систему потенційно менш гнучкою, а також знижує її пропускну здатність та сповільнює. Як працює система можна детально побачити на діаграмі, яка зображена на рисунку 1.4.

Пропозиції щодо автоматизації бізнес-процесу

Після аналізу процесу формування маршрутів диспетчерами можна виявити деякі недоліки. Головний з яких – це відсутність автоматизації процесів, що б покращило якість отримання послуги та фінансові показники компанії. Також відсутність механізму регулювання кількості виконаної роботи між водіями, зараз можлива ситуація в якій зміна одного водія виявляється набагато довшою за зміну іншого. Також проблемним місцем є процес додавання нових об'єктів з якими укладаються контракти та внесення їх в систему, адже на даний момент це потребує втручання диспетчера, що не є ефективним.

Для покращення роботи компанії запропоновано розробити АС для автоматизації бізнес-процесів, яка б дозволяла покращити якість надання послуг, скоротила час їх надання, гарантувала більш справедливі умови праці для водіїв, щоб вони мали можливість закінчувати зміну одночасно. Як наслідок всіх цих автоматизацій ми отримаємо систему, яка дасть змогу покращити фінансові показники компанії.



Рисунок 1.4 – Процес формування маршруту до автоматизації

Після автоматизації

Адміністратором створюються облікові записи для всіх учасників процесу, та додаються всі об'єкти в систему. На цьому етапі будуються маршрути між кожною парою точок зареєстрованих в системі, що в подальшому дасть можливість будувати маршрути динамічно та повністю вилучити людський фактор на етапі складання та планування маршруту. Після цього локальні менеджери починають залишати заявки на вивіз сміття.

Далі система автоматично запускає планувальник, який буде розраховувати всі маршрути. На цьому етапі будуть виконуватися оптимізації, які допоможуть гарантувати приблизно одночасне закінчення зміни для всіх водіїв. Досягтися це буде шляхом того, що для тих водіїв, які працюють на більш віддалених від місця відвантаження сміття будуть заплановані рейси з меншою кількістю об'єктів, а для тих, які працюють ближче до місця відвантаження навпаки — рейси з більшою кількістю об'єктів.

Після того як початкові маршрути обраховані, водії починають їх опрацьовувати (все відбувається автоматично, без участі людини). Коли хтось із водіїв повністю заповнює свій автомобіль, він відправляється на базу, щоб відвантажитися, і в цей момент він звітує про це в систему. Цей момент для системи є сигналом про те, що необхідно перевірити, чи ніхто з водіїв не почав відставати. У такому разі система починає автоматично перераховувати маршрути для всіх водіїв.

Головною перевагою даної системи після автоматизації стає досягнення справедливого розподілу праці між водіями. Більше не буде відбуватися ситуація, коли один водій змушений працювати набагато довше ніж інший, через різні причини, які можливо навіть не залежать від нього.

Те як процес буде відбуватися після автоматизації можна побачити на діаграмі зображеній на рисунку 1.5.



Рисунок 1.5 – Процес формування маршруту після автоматизації

Як ми можемо побачити, після автоматизації диспетчер повинен втручатися тільки в тих випадках, коли система починає працювати некоректно, тобто він має просто наглядати за нею.

1.2 Опис постановки задачі

Система призначена для автоматизації процесів діяльності спеціалізованої транспортної компанії.

Метою функціонування системи є покращення показників часу обслуговування, а також забезпечення максимально рівномірного розподілу обсягу виконаної роботи за зміну між працівниками.

Покращення цих показників планується досягти за рахунок розробки кращого алгоритму складання розкладу перевезень, який окрім таких стандартних факторів оптимізації таких як час та відстань, також буде враховувати фактор того, наскільки рівномірно робота в межах зміни розподілена всіма працівниками.

Для досягнення поставленої мети необхідно реалізувати наступні задачі:

- ведення інформації про клієнта (від внесення в систему до отримання кінцевих звітів стосовно виконаної роботи);
- ведення інформації про місця дислокації клієнта (додавання до бази всіх контейнерів клієнтів, а також побудова повної матриці відстаней між усіма наявними об'єктами для того, щоб мати змогу будувати маршрути між кожною парою з них);
- формування статистичних звітів (звіти з інформацією про час та відстань, яку проїхало кожне авто, а також сумарний звіт-підсумок по усім рейсам, які були виконані під час зміни, сумарна статистика за будь-який проміжок часу та порівняльний аналіз цих звітів);
- ведення архіву наданих послуг (збереження історії всіх подій, що відбуваються в системі та їх поточний статус);
- редагування на мапі точного місця дислокації за GPS координатами (можливість для водіїв інформувати систему про стан свого поточного положення);

- формування сповіщення про надані послуги транспортною компанією (після того, як заявки були опрацьовані, замовник може бути сповіщеним про цю подію);
- перегляд історії обслуговування за визначеними параметрами (переглядати історію роботи системи за різними критеріями такими як: час, координати, конкретний транспортний засіб);
- затвердження обсягу наданих послуг (динамічне формування обсягу роботи на поточну зміну базуючись на заявках, які були прийняті та визначені як ті, що мають бути опрацьовані в першу чергу);
- ведення графіку обслуговування об'єктів дислокації клієнтів (прогнозування часу обробки заявки, яке буде надано кінцевому користувачеві);
- формування денного завдання для водія (динамічний розрахунок маршруту на зміну для водія, а також орієнтовний час його виконання);
- подання звітності з фактичного обсягу роботи (звітування щодо кожного опрацьованого об'єкта, фактичного місця знаходження, а також остаточного стану виконання завдання на зміну. За результатами підрахунків цих даних можуть бути зроблені висновки стосовно ефективності кожного з водіїв та прийняті рішення щодо матеріальних заохочень).

Для цього мають бути вирішені такі задачі:

- побудова спеціалізованого логістичного алгоритму (алгоритм має враховувати одразу всі необхідні для оптимізації параметри);
- побудова спеціалізованого кластерного алгоритму;
- побудова алгоритму гарантування максимально одночасного завершення зміни для всіх водіїв (основною задачею оптимізації є забезпечення максимально справедливого та рівномірного розподілу роботи між бригадами, що дуже ускладнюється недовизначеністю показників місткості автомобіля, що ускладнює планування та погіршує його точність).

1.3 Огляд аналогів системи, що розробляється

У світі існує багато систем, що вміщують в собі моніторингову систему (відображення координат автотранспорту підприємства на мапі), але майже не існує

систем, що містять в собі як моніторингову частину, так і логістичну. Логістична структура здебільшого виноситься в окрему розробку, та більшість з них це «хмарна» технологія, що не завжди підходить для клієнта з причин комерційної таємниці.

Також не існує систем, що вміщують в собі складові, які одночасно поєднують менеджера, диспетчера, клієнта та водія. Системи, що не мають такого функціоналу зазвичай зменшують продуктивність роботи та рівень задоволення клієнтів і перевізника. З появою мережі інтернет та загальною комп'ютеризацією суспільства створились умови для виникнення подібних програм.

Автоматизація всіх вищесказаних процесів та кабінетів дозволить більш оптимально надавати послуги клієнту, скоротити час на усунення проблем у взаємодії між усіма ланками компанії та клієнтом, наглядність для водія про його маршрут і т.д.

У перевізника також виникає потреба оптимізувати свій маршрут та відобразити його на мапі як для диспетчера так і для водія. Давайте розглянемо більш детально відомі системи з моніторингу:

- Tracker Online [1], що має наступний функціонал:
 - 1) хмарна технологія доступу;
 - 2) відображення місцеперебування та стану датчиків в реальному часі;
 - 3) велика кількість постійно оновлюваних карт (GoogleMaps [2,3], OpenStreetMap [4], Visicom [5], Yandex [6]);
 - 4) відображення інформації про завантаженість доріг (пробки);
 - 5) контроль геозон;
- BENISH GPS [7] – функціонал системи майже схожий з попередньою системою з єдиною різницею, що сама компанія розробник тісно співпрацює з структурами МВС;
- Wialon [8] – дана система найбільш розповсюджена та має в собі наступні переваги порівняно з попередніми:
 - 1) має версію як «хмарну» так і серверну;
 - 2) має можливості написання своїх додатків до системи через SDK системи;
 - 3) має компонентну структуру.

- Ant Logistics – система дозволяє керувати набором об’єктів на карті, які відносяться до акаунту користувача, а також будувати між ними маршрути.

Мінусом даних систем є:

- окрема абонплата на дані системи;
- відсутність системи в якій одночасно можна відслідковувати координати транспортних засобів, а також будувати маршрути;
- орієнтованість на доставку.

Оглянувши дані системи та пропоновані логістичні складові можемо прийти до висновку що жодна з них не відповідає тому функціоналу, що ставиться або ж має надлишкову орієнтованість під доставку.

1.4 Рішення з інформаційного забезпечення

Дані будуть зберігатися в реляційній базі даних PostgreSQL [9, 10, 11]. Специфіка полягає в тому, що для вирішення даної задачі необхідно зберігати матрицю відстаней, на даний момент для цього була адаптована реляційна база даних, але вона не є оптимальним рішенням для подібного роду задач, через те що, для додавання кожного нового запису, необхідно додати ще $N+1$, що в свою чергу призводить до надзвичайного розростання кількості записів, наприклад для зберігання матриці розміром всього 1000 на 1000 необхідно зберігати 1000000 записів. Також оскільки на момент отримання результатів запиту реляційні бази віддають лінійні структури, то необхідно проводити агрегації багатомільйонних масивів на сервері, що надзвичайно погано відображається на загальній швидкості роботи системи та кількості ресурсів які вона використовує.

Враховуючи приведені вище недоліки в подальших планах є зміна база даних для збереження саме цієї частини даних на щось, що може так само забезпечити зручну вибірку даних як SQL запити, проте дозволить значно швидше виконувати операції додавання та видалення рядків та стовпців з матриці та й загалом буде зберігати матрицю в більш природному (квадратному вигляді). Попередньо проведені аналізи показують, що для подібного роду задач максимально підходять так звані “колончасті” бази даних.

Вхідні дані

Визначимо вхідні дані які необхідно надати системі для того, щоб з нею можна було почати працювати:

- набір об'єктів які необхідно опрацювати – задається парою географічних координат;
- матриця відстаней між об'єктами;
- дані автопарку;
- інформація про робочі зміни;
- параметри алгоритму складання розкладу перевезень;
- статистичні дані стосовно середнього значення показників місткості сміттєзбирального автомобіля.

Вихідні дані

Як результат система видає наступні дані:

- набір маршрутів для кожного транспортного засобу;
- покрокова історія обробки зміни;
- детальна статистика стосовно кожної з бригад, яка брала участь в зміні;
- підсумовану статистику щодо роботи всієї зміни.

Формат збереження даних

У додатку А на плакаті 2 наведена схема даних модулю, який відповідає за логістичну складову.

Схема даних містить велику кількість посилань, які мають тип один до багатьох та багато до багатьох, детальну інформацію про посилання наведемо в таблицях 1.1 та 1.2 відповідно.

Таблиця 1.1 – Пояснення зв'язків типу один до багатьох

Батьківська таблиця	Батьківська колонка	Дочірна таблиця	Дочірна колонка	Пояснення
Day	Id	Route	Day_id	Відношення відбувається за допомогою колонок <i>route.day_id</i> яка посиляється на <i>day.id</i> , відношення визначає які маршрути відносяться до конкретного дня
Car	Id	Route	Car_id	Відношення відбувається за допомогою колонок <i>route.car_id</i> яка посиляється на <i>car.id</i> , відношення визначає які автомобілі приймали участь в конкретній зміні

Таблиця 1.2 – Пояснення зв'язків типу багато до багатьох

Таблиця 1	Таблиця 2	Зв'язуюча таблиця	Пояснення
Point	Point	Distance	Відношення в якому таблиця <i>point</i> посиляється сама на себе за допомогою допоміжної таблиці <i>distance</i> , яка визначає зв'язок на перетині двох зовнішніх ключів (<i>destination_point_id</i> та <i>source_point_id</i>) зберігає також інформацію про відстань між точками, подібна організація структури бази даних дає змогу контролювати той факт, що відстань з одного пункт до іншого, та назад може бути різною
Point	Day	Day_point	Відношення реалізовано за допомогою зв'язуючої таблиці <i>day_point</i> яка має посилання на 2 батьківські таблиці (<i>day_point.day_id</i> -> <i>day.id</i> та <i>day_point.point_id</i> -> <i>point.id</i>), зберігає інформацію стосовно того які об'єкти відносяться до якого дня

Продовження таблиці 1.2

Таблиця 1	Таблиця 2	Зв'язуюча таблиця	Пояснення
Route	Point	Route_point	Відношення реалізовано за допомогою зв'язуючої таблиці route_point яка має посилання на 2 батьківські таблиці (route_point.route_id -> route.id та route_point.point_id -> point.id), зберігає інформацію стосовно того які точки відносяться до якого маршруту
Day	Car	Day_car	Відношення реалізовано за допомогою зв'язуючої таблиці day_car яка має посилання на 2 батьківські таблиці (day_car.car_id -> car.id та day_car.day_id -> day.id), зберігає інформацію стосовно того які автомобілі працювали в конкретну зміну

Схема даних складається з наступних таблиць:

Point – таблиця для збереження об'єктів на карті. Значення колонок наведені у таблиці 1.3.

Таблиця 1.3 – Призначення колонок таблиці Point

Назва колонки	Пояснення
Id	Унікальний ідентифікатор об'єкту на карті
X	Географічна широта, задається числом в діапазоні від -180 до 180 включно
Y	Географічна довгота, задається числом в діапазоні від -90 до 90 включно
Is_base	Чи являється дана точка базовою
Is_full	Чи являється дана точка заповненою в даний момент

Day – таблиця для збереження загальних даних про робочу зміну. Значення колонок наведені у таблиці 1.4.

Таблиця 1.4 – Призначення колонок таблиці Day

Назва колонки	Пояснення
Id	Унікальний ідентифікатор поточного робочого дня
Car_capacity	Мінімальна кількість контейнерів які вміщує в собі автомобіль
Car_capacity_error	Максимальна похибка на скільки відсотків місткість автомобілю може відрізнятись від очікуваної, задається значеннями від 0 до 1
Completed	Поточний стан даної зміни (у випадку коли зміна була тільки запланована або перебуває в процесі обробки - false, коли зміна успішно завершена - true)
Date	Дата яка відповідає дню
End_time	Час завершення обробки дню
Start_time	Час початку обробки дню
Total_distance	Загальна відстань яка була подолана сумарно всіма ТЗ які брали участь в цій зміні (в кілометрах)
Base_point_id	Ідентифікатор базової точки (звалища) на карті, куди всі ТЗ після наповнення їздять відвантажуватися

Car – таблиця в якій зберігаються дані про транспортні засоби (ТЗ) компанії. Значення колонок наведені у таблиці 1.5.

Таблиця 1.5 – Призначення колонок таблиці Car

Назва колонки	Пояснення
Id	Унікальний ідентифікатор ТЗ
Busy	Стан в якому ТЗ знаходиться в даний момент (бере участь в зміні на даний момент чи ні)

Distance – таблиця для збереження даних про відстані між об’єктами. Значення колонок наведені у таблиці 1.6.

Таблиця 1.6 – Призначення колонок таблиці Distance

Назва колонки	Пояснення
Id	Унікальний ідентифікатор запису про відстань
Distance	Відстань між точками в км
Destination_point_id	Ідентифікатор кінцевої точки
Source_point_id	Ідентифікатор початкової точки

Route – таблиця для збереження загальної інформації про маршрут. Значення колонок наведені у таблиці 1.7.

Таблиця 1.7 – Призначення колонок таблиці Route

Назва колонки	Пояснення
Id	Унікальний ідентифікатор маршруту
Distance	Загальна відстань маршруту
End_time	Час завершення обробки маршруту
Start_time	Час початку обробки маршруту
Car_id	Ідентифікатор автомобілю який обслуговував маршрут
Day_id	Ідентифікатор дня в який було опрацьовано маршрут

Day_car – зв’язуюча таблиця для збереження інформації стосовно ТЗ, які брали участь в робочій зміні. Значення колонок наведені у таблиці 1.8

Таблиця 1.8 – Призначення колонок таблиці Day_car

Назва колонки	Пояснення
Car_id	Ідентифікатор ТЗ
Day_id	Ідентифікатор робочої зміни

Day_point – зв’язуюча таблиця для збереження інформації стосовно об’єктів, які були опрацьовані в межах робочої зміни. Значення колонок наведені у таблиці 1.9.

Таблиця 1.9 – Призначення колонок таблиці Day_point

Назва колонки	Пояснення
Point_id	Ідентифікатор об’єкту
Day_id	Ідентифікатор робочої зміни

Route_point – зв’язуюча таблиця для збереження інформації стосовно об’єктів, які були опрацьовані в межах маршруту. Значення колонок наведені у таблиці 1.10.

Таблиця 1.10 – Призначення колонок таблиці Route_point

Назва колонки	Пояснення
Point_id	Ідентифікатор об’єкту
Route_id	Ідентифікатор маршруту.

Висновок до розділу

Метою розділу є збір даних про принципи функціонування реальних підприємств, які займаються вивезенням сміття. На основі отриманих даних було сформовано вимоги до процесу автоматизації складання розкладу для подібного роду перевезень.

На основі аналізу виділено основні напрями розробки системи. Було проведено аналіз вже чинних систем та виділено сильні та слабкі сторони кожної з них. Основний недолік який було виявлено, це те, що на даний момент в аналогічних системах відсутня оптимізація, яка б гарантувала максимально справедливий розподіл роботи між бригадами в межах однієї зміни.

Проведено реінжиніринг процесів, який дозволить надавати нові та більш якісні послуги, привабити нових клієнтів. Взаємодія всіх учасників процесу через АС скоротить час реагування, якість надання послуг та фінансові витрати на зв'язок. Буде скорочено час обробки кожної зміни, що позитивно зобразиться як на рівні задоволеності клієнтів, так і на рівні задоволеності працівників компанії, які зацікавлені в тому, щоб зміни були рівними по тривалості для всіх.

Буде скорочено час на додавання нових об'єктів обслуговування тому, що з цього етапу буде вилучено людський фактор. З'явиться можливість повністю автоматично будувати динамічні маршрути та динамічно розраховувати зміни, що прямим чином задовольняє поточні потреби компаній такого роду. Також такий динамічний підхід дозволить більш гнучке реагування на різні нестандартні ситуації, наприклад вихід з ладу транспортних засобів або зміни з аномально високою, чи низькою завантаженістю.

Після реінжинірингу показники компанії з якості обслуговування та фінансові показники покращаться. Це все відобразиться на рівні соціального задоволення мешканців, адже сміття завдяки використанню даної системи буде вивозитися частіше.

З урахуванням усіх вимог які має розглянута задача оптимізації була сформована архітектура для майбутньої системи, а саме клієнт-серверна. Подібна архітектура була обрана через те, що існує велика кількість виконавців для яких потрібен єдиний центр управління та координації їх дій, а саме – сервер.

Виділено головних акторів системи та визначено функції кожного з них. Встановлено взаємозв'язки між акторами.

Було обрано тип бази даних, а саме – реляційна, а також проаналізовано її сильні та слабкі сторони для даного варіанту використання. Також була розроблена схема бази даних.

2 МОДЕЛІ ТА МЕТОДИ ПІДТРИМКИ ПРОЦЕСІВ ДІЯЛЬНОСТІ СПЕЦІАЛІЗОВАНОЇ ТРАНСПОРТНОЇ КОМПАНІЇ

2.1 Змістовна постановка задачі

У сучасному світі існує велика кількість компаній, які у своїй діяльності стикаються з завданням перевезення різного виду вантажів. Це визначило великий попит на ефективне вирішення завдань логістики для різноманітних умов виконання перевезень та стимулювало розвиток точних та наближених методів вирішення таких завдань. Проте кожна з галузей перевезень має окрім типових особистісних складові, що роблять наявні підходи недостатньо ефективними. Таким чином класична задача маршрутизації транспортних засобів з розділеним обслуговуванням (SDVRP) в кожному випадку має свої особливості.

Для обраного нами процесу діяльності щодо вивозу сміття для утилізації фактичні показники як ваги, так і об'єму завантаженості баків (далі – Вантажних Одиниць (ВО)) замовника є суттєво недовизначеним. Над темою недовизначеності показників вже працювали такі науковці як: Нариньяні А. С. у своїй роботі “Недовизначеність в системах представлення та обробки знань” [12], Fortin J. У роботі “Gradual Numbers and their Application to Fuzzy Interval Analysis” [13] та Эндрю А. у роботі “Штучний інтелект” [14].

Єдиною інформацією, якою можна скористатися для оцінювання, є дані про загальну множину ВО, множину ВО що були фактично обслуговані за конкретний рейс транспортним засобом (далі ТЗ), вантажність та максимальний об'єм вантажу для ТЗ. Для інших показників наявні лише дуже усереднені, з великим розбігом значень, оцінки.

Для врахування всіх специфічних факторів які має ця задача необхідно виконати комбінування одразу декількох типів алгоритмів, а саме побудова найкоротшого шляху під час обходу всіх контейнерів та повернення в початкову точку, що приводить нас до класичної задачі комівояжера [15, 16, 17], групування об'єктів та розподіл між виконавцями (конкретними транспортними засобами які будуть опрацьовувати саме цей об'єкт), а також оптимізація, яка б гарантувала максимально наближений час завершення опрацювання для кожного ТЗ. Розглянемо

причини застосування та значення, які кожна з цих груп алгоритмів здатна вирішити або оптимізувати в нашому конкретному випадку більш детально.

2.2 Огляд методів розв'язання

2.2.1 Методи розв'язання задачі комівояжера

Задача комівояжера у своєму найбільш відомому вигляді має наступне формулювання “Задача комівояжера полягає у знаходженні найвигіднішого маршруту, що проходить через вказані міста хоча б по одному разу. В умовах завдання вказуються критерій вигідності маршруту (найкоротший, найдешевший, сукупний критерій тощо) і відповідні матриці відстаней, вартості тощо. Зазвичай задано, що маршрут повинен проходити через кожне місто тільки один раз, в такому випадку розв'язок знаходиться серед гамільтонових циклів” [18, 19]. На жаль, ця задача відноситься до NP [20, 21, 22] повних, що в свою чергу означає, що в даний момент не існує алгоритмів які б могли гарантувати точний розв'язок, окрім повного перебору, що є прийнятним лише для задач надзвичайно малої розмірності, адже кількість унікальних маршрутів розраховується за формулою $\frac{(n-1)!}{2}$, що в свою чергу вже при п'ятнадцяти містах становить неймовірно $\frac{(15-1)!}{2} = 43589145600$ варіантів, саме тому всі алгоритми які використовуються на даний момент та вважаються прийнятними є евристичними, тобто такими, що знаходять на оптимальний, а наближено оптимальний маршрут, що в свою чергу дає їм можливість виступати в ролі певного компромісу між часом виконання та точністю.

Оскільки в конкретній прикладній задачі (вивезення сміття у великому місті) кількість об'єктів може сягати десятків тисяч на добу, що автоматично зміщує пріоритети при обиранні компромісу між точністю та швидкістю обчислень, саме до другого варіанту.

Імітація відпалу

Алгоритм імітації відпалу [26, 27, 28] — це загальний алгоритмічний метод розв'язання задачі глобальної оптимізації, насамперед дискретної, а також комбінаторної оптимізації тому, що в них пошук глобального розв'язку базується на

наслідуванні фізичного процесу, що відбувається при кристалізації речовини, в тому числі при відпалі металів. Завдяки тому, що алгоритм створений для вирішення задачі глобальної оптимізації він також може бути використаний для вирішення задачі комівояжера.

Алгоритм можна використати для того щоб згенерувати послідовність рішень задачі комбінаторної оптимізації, якщо провести певні аналогії між фізичною системою багатьох частинок і завданням комбінаторної оптимізації, а саме:

- розв'язки задачі комбінаторної оптимізації еквівалентні станам фізичної системи;
- вартість рішення еквівалентна енергії стану системи.

Для розв'язання задачі необхідно мати певний керуючий параметр, який буде зображати ефективність кожного нового розв'язку, у фізичній імплементації роль цього параметру відіграє температура. Визначимо головні етапи з яких складається процес імітації відпалу:

- ініціалізація під час якої обчислюються початкові значення всіх параметрів та задається початковий стан системи;
- сусідній розв'язок обирається під час генерації нового стану системи, відбувається порівняння його ефективності відносно поточного розв'язку, головною його особливістю алгоритму є те, що з певною долею ймовірності він може обрати гірше рішення, а робиться це для того, щоб уникнути хибних ступорів в точках локального оптимуму;
- обчислення характеристик отриманого розв'язку, таких як довжина та ймовірності з якою даний розв'язок буде прийнятий як оптимальний.

Після багаторазового повторного виконання описаних вище кроків, отримуємо послідовності однорідних ланцюжків Маркова обмеженої довжини. У випадку, коли значення керуючого параметра стає нижчим за мінімум, який був заданий на моменті ініціалізації, ми отримаємо підсумковий результат моделювання проведеного методом відпалу.

Оптимізація мурашиної колонії

Мурашиний алгоритм [29, 30, 31] — один з ефективних поліноміальних алгоритмів для знаходження наближених розв'язків задачі комівояжера, а також аналогічних завдань пошуку маршрутів на графах.

Головна ідея алгоритму полягає в імітації поведінки мурашок в межах їх колонії. Використовується саме той підхід, який мурашки виробили за мільйони років еволюції для орієнтування в умовах майже повної відсутності світла та звуку. Вони навчилися відкладати феромони які здатні відчувати завдяки своїм органам чуття. Маршрути які є вдалимими – позначаються великою кількістю феромонів, навдалі – навпаки.

Робота алгоритму починається з розміщення мурашок у вершинах графу, а напрямок їх руху визначається ймовірнісним методом. Результат алгоритму не є гарантовано оптимальним і точним, в найгіршому випадку існує ймовірність, що він може виявитися навіть одним із гірших, проте велика кількість повторень в кінцевому результаті видає досить точний результат.

Зробимо припущення, що навколишнє середовище для мурах представляє повний зв'язний неорієнтований граф, тобто мураха може перейти з кожної точки в іншу та назад. Кожне ребро має вагу, яка відображає відстань між двома вершинами, що ним з'єднується. Граф є двонаправленим, тому мураха може подорожувати по грані в будь-якому напрямку.

Ймовірність включення ребра в маршрут окремої мурахи визначається пропорційно до кількості феромонів на цьому ребрі, а кількість відкладеного феромону пропорційна до довжини маршруту. Оскільки кількість феромону – це обмежений ресурс, то чим коротший маршрут, тим більше феромону буде відкладено на його ребрах. З цього слідує, більша кількість мурах буде включати ребро з великою кількістю феромонів в синтез власних маршрутів.

Якщо покладатися лише вибір ребер з найбільшою кількістю феромонів, то подібне моделювання дуже швидко спричинить передчасну збіжність маршруту, тобто більшість мурашок почне рухатися по локально-оптимальному маршруту.

Для уникнення подібного ефекту потрібно ввести моделювання від'ємного зворотного зв'язку у вигляді випаровування феромону. Слід враховувати, що у

випадку занадто швидкого випаровування феромону, ця оптимізація може призвести до втрати пам'яті колонії і забування хороших рішень, з іншого боку, збільшення часу випарів може призвести до отримання стійкого локального оптимального рішення.

2.2.3 Методи розв'язання задачі кластеризації

Задля можливості подальшого масштабування виникла необхідність розділення об'єктів на кластери [23, 24, 25], у випадку даної прикладної задачі кластери можуть відповідати районам міста. Алгоритм передбачає вирішення задачі комівояжера для всіх точок кожного кластеру, у випадку коли алгоритми кластерування не використовуються (що означає, що всі об'єкти входять в один кластер) будь-яке прийнятне вирішення задачі комівояжера за прийнятний час стає фактично неможливим, адже вже для 100 об'єктів загальна кількість унікальних шляхів становить

466631077219720763408496194281333502453579841321908107342964819476087999
966149578044707319880782591431268489604136118791255926054584320000000000
000000000000

Але проаналізувавши задачу, було виявлено, що жодний з існуючих алгоритмів кластеризації не може бути використаний в даному випадку, адже всі вони направлені лише на наступні типи оптимізації:

- які генерують довільну кількість кластерів, та базуються лише на щільності об'єктів;
- аналогічні до попереднього пункту, але з фіксованим значенням кількості кластерів;
- алгоритми з фіксованою кількістю кластерів а також рівною кількістю об'єктів в кожному з кластерів;
- алгоритми з фіксованою кількістю кластерів та максимально рівною відстанню яка буде обрахована задачею комівояжера запущеною для точок кластера.

2.2.3 Методи розв'язання задачі планування розкладу перевезень

На даний момент, тенденції розвитку світового бізнесу націлені на гуманізм в сторону працівників свої підприємств, що в нашому прикладному випадку

виражається в тому, що компанії зацікавлені в тому, щоб всі їх бригади, що закріплені за кожним конкретним ТЗ виконували однакову кількість роботи та не траплялося ситуацій коли комусь доводиться працювати значно довше через те, що йому дістався більш віддалений маршрут. Саме тому виникла необхідність оптимізувати побудову маршрутів яка б враховувала ще й цей критерій. Очевидно, що подібного роду оптимізації можуть негативно сказуватися на інших параметрах, таких як відстань, саме тому необхідно дуже обережно контролювати те, щоб покращення одного параметру не призводило до погіршення інших.

Класичний підхід

Класичний підхід до вирішення задачі складання розкладу перевезень з недовизначеними показниками заключається в тому, що сам по собі фактор недовизначеності просто ігнорується, тобто всі маршрути розраховуються та виконуються з урахуванням певного середнього значення об'єму транспортного засобу, що в свою чергу під час реального виконання може вилитися в дуже не оптимальні результати через те, що у випадку коли об'єм виявиться більше запланованого, то є ймовірність того, що буде необхідність робити ще один рейс заради дуже незначної кількості контейнерів які необхідно опрацювати, внаслідок чого транспортний засіб впродовж усього маршруту буде майже порожнім, що є економічно не вигідним та значно збільшує витрати.

Як один із способів вирішення цієї проблеми – це варіант залишати ці останні контейнери на наступний день виставляючи їм при цьому певні пріоритети які б гарантували, що сьогодні цей об'єкт буде гарантовано опрацьований. Проте чи готові на такі компроміси мешканці та й самі компанії які надають подібного роду послуги залежить від їх бізнес моделі, та контрактів які були укладені з містом.

Проте, такий підхід в будь-якому разі не вирішує проблему однакової довжини зміни для всіх бригад, оскільки рівномірний розподіл контейнерів за їх кількістю ніяк не враховує фактор віддаленості сектору роботи одного транспортного засобу в порівнянні до іншого, так наприклад може виявитися що для автомобіля який працює максимально близько до звалища зробити 2 рейси буде швидше ніж 1 для автомобіля який працює на максимально віддалених ділянках, при чому чим більше рейсів кожна

бригада робить за зміну, тим сильніше цей фактор буде впливати на збільшення розриву в часі виконання.

Метод динамічного перерозподілу

Головна ідея цього методу полягає в тому, що з самого початку він починає працювати абсолютно аналогічно до класичного, а основна відмінність полягає в тому, що під час кожного завершення маршруту будь-яким з транспортних засобів які приймають участь у зміні або в момент коли бригада звітує про завершення обробки всіх об'єктів які були для неї виділені відбувається перевірка чи не відбулося відставання певної бригади від всіх інших (базується подібна перевірка на кількості контейнерів які залишилися опрацювати кожній з бригад), після того як кількість контейнерів які залишилися для кожного з автомобілів було визначено, підраховується відсоток на який автомобіль з мінімальним значення випереджає автомобіль з максимальним. Якщо цей відсоток перевищує певний поріг який задається конфігурацією відбувається перерахунок.

Перерахунок може здійснюватися багатьма способами, найпростіший з яких це агрегувати всі контейнери які залишилися, та виконати початкову стадію алгоритму з самого початку але лише для цих об'єктів, а після підрахунку переназначити задачі для кожного з виконавців.

Цей метод гарно справляється з задачею забезпечення рівномірної кількості роботи між усіма виконавцями, але він має суттєвий недолік який заключається в тому, що маршрути які були перераховані динамічно виявляються значно менш оптимальними ніж ті які були заплановані з самого початку, через що значно збільшується відстань яку транспортним засобам необхідно подолати, що в свою чергу збільшує витрати на паливо та мінімізує користь подібної оптимізації.

Подібний метод може давати позитивний результат лише у випадку коли число рейсів які здійснює кожен з автомобілів є досить великим, і чим більше буде це число, тим кращі результати будуть в порівнянні з класичним алгоритмом, але в нашій конкретній прикладній задачі число рейсів за зміну ніколи не перевищує діапазон від 3х до 5ти чого не є достатнім для того, щоб ця оптимізація дала відчутній вииграш у часі в порівнянні з класичним методом.

Метод рівномірно розподілу роботи на етапі планування

Цей метод відрізняється від класично лише на етапі планування, а саме під час підрахунку кластерів має бути виконана попередня симуляція зміни з використанням середньо-статистичних значень для недовизначених показників які були отримані з статистичного аналізу попередніх змін (чим більше факторів таких як температура навколишнього середовища, сезон, вологість буде враховано, тим точнішим виявиться фінальний прогноз).

Базуючись на цих даних після проведення симуляції ми отримуємо наближений розрахунок того як цю зміну б виконував 1 автомобіль (без урахування часу), що дає нам можливість визначити яку кількість кілометрів в середньому має пройти кожен транспортних засобів який приймає участь в зміні. Після цього ми ітеративно намагаємося розподілити кількість кілометрів порівну між водіями.

Перевагою цього методу є те, що маршрути якими рухаються транспортні засоби розраховані з самого початку, що робить їх умовно максимально оптимальними наскільки це можливо, тобто ми зберігаємо головну перевагу класичного методу, тобто відстань, також ми отримуємо приблизно рівний час завершення зміни для всіх бригад, що є головною перевагою динамічного методу, тобто цей підхід увібрав в себе найкращі риси від двох попередніх.

Проте він також має свої недоліки, найголовніший з яких заключається в тому, що він не гнучкий до динамічних змін обставин, а саме у випадку якщо якийсь транспортний засіб вийде з ладу, алгоритм не буде здатний автоматично замінити його на інший.

Як було виявлено однією з головних вимог серед подібних спеціалізованих компаній які займаються вивезенням сміття та потребують автоматично складання розкладу перевезень є необхідність забезпечення максимально наближеного часу завершення для всіх бригад які виходять на зміну, окрім цього не варто забувати про два основних критерії мінімізації будь-якої транспортної задачі, а саме мінімізація загального часу обробки та відстані (що еквівалентно мінімізації витрат на паливо).

Використовуючи різні значення для параметрів оптимізації можна підлаштувати їх під різні бізнес моделі (оскільки здебільшого покращення одного параметру призводить до погіршення одного чи одразу двох інших).

В нашому випадку головною ціллю оптимізації є мінімізація розриву в часі виконання для всіх бригад, при цьому з мінімальним впливом на показники загальної відстані. Очевидно, що з точки зору відстані, краще себе будуть показувати маршрути які з самого початку гарно заплановані (тобто такі які не враховують недовизначеність показників об'єму), з цього стає зрозуміло, що фінальна версія алгоритму повинна з самого початку провести повне планування маршрутів, щоб в подальшому уникнути динамічного перерахунку, що може призвести до погіршення показників з оптимальності маршруту, а також при цьому з самого початку розділити роботу максимально порівну з точки зору часу виконання.

2.3 Огляд існуючих рішень

Було проведено пошук та аналіз наукових праць схожих за змістом, серед розглянутих робіт найбільш близькі за містом виявились наступні:

- “Інформаційні технології в розв’язанні транспортних задач” [32] – дана робота розглядає дуже детально розглядає різноманітні види математичних моделей для задач організації та планування транспортних моделей, серед яких найбільш близькою за змістом до задачі яка розглядається в межах поточної дисертації є “Задача про розподіл транспортних засобів за витратами часу”;
- “Розробка графіка руху транспортних засобів при організації вантажних перевезень” [33] – дана робота розглядає наскільки правильно сплановані графіки руху впливають на загальну ефективність транспортних перевезень, враховуючи навіть такі фактори як психологічний стан водіїв. Також в цій роботі розглянуто методи розробки графіків руху а також проведено оцінку їх ефективності;
- “Моделювання процесу планування вантажоперевезень” [34] – в даній роботі побудовано та досліджено базовий варіант моделі процесу планування вантажоперевезень, що складається з послідовності обраних окремих підзадач, утворених відібраною системою спрощувальних припущень, що дає змогу розв’язати поставлену задачу в умовах багатокритеріальності та наявності часових обмежень;

- “Основи дослідження операцій у транспортних системах: приклади та задачі” [35] – в даній роботі викладено теоретичні відомості щодо побудови математичних моделей та розв’язання найбільш поширених оптимізаційних задач, спрямованих на удосконалення транспортних процесів. Для розв’язання подібного класу задач розглянуто наступні методи дослідження операцій: лінійне програмування, динамічне програмування, сітьове планування та управління, теорія масового обслуговування, імітаційне моделювання;
- “Методика вирішення завдань пошуку оптимальних туристичних маршрутів алгоритмами наслідування мурашиної колонії” [36] – дана робота розглядає методику вирішення задач пошуку оптимальних маршрутів туризму. В цій роботі здійснено формальний математичний опис задачі проектування маршрутної туристичної мережі а також проведено аналіз чисельних методів їх вирішення. Як висновок, в цій роботі було виявлено, що одним з найбільш перспективних методів для знаходження маршрутів на сьогоднішній день є метод оптимізації наслідуванням мурашиної колонії. Також приведено приклад, який показує, як в алгоритми вирішення прикладних дискретних задач оптимізації можливо впровадити складові самоорганізації мурах.

2.4 Розв’язання транспортної задачі з рівномірним розподілом роботи

В межах цього розділу сформулюємо математичну модель задачі, дослідимо властивості математичної моделі а також розпишемо всі алгоритми які були використані при розв’язанні:

- алгоритм імітації відпалу для вирішення задачі комівояжера;
- генетичний алгоритм імітування мурашиної колонії для вирішення задачі комівояжера;
- алгоритм кластеризації для формування планів зміни.

2.4.1 Постановка задачі

Дано:

- c – середнє значення ємності вантажного автомобіля виміряне в кількості контейнерів ;
- n – кількість об'єктів які необхідно опрацювати;
- m – кількість транспортних засобів;
- b – точка з координатами відправного та кінцевого пункту;
- D – матриця відстаней розмірності $(n + 1) \times (n + 1)$ (одна додаткова точка, це точка b), де порядкові номери стовпцю означає точку відправлення, порядковий номер колонки – точку призначення, а саме значення – відстань;
- v – середня швидкість транспортних засобів;
- t – час початку опрацювання.

Побудуємо за допомогою алгоритму комівояжера маршрут обходу всіх точок включаючи точку $b - R_{n+1}$. Визначимо довжину маршруту – L , у випадку, якщо б його обслуговував лише 1 автомобіль. Для цього скористаємося формулою:

$$L = \sum_{i=1}^{n+1} (D_{R_i R_{i+1}} + (D_{R_i R_{i+1}} + D_{R_i R_{i+1}})[i \bmod c = 0]) \quad (2.1)$$

Після цього визначимо середнє значення відстані для кожного автомобіля – d :

$$d = \frac{L}{m} \quad (2.2)$$

Далі базуючись на формулі (2.1) та (2.2) побудуємо очікувані маршрути для кожного з транспортних засобів – R_j , на початку $j = 1$:

$$R_j = \sum_{i=1}^{n+1} ((D_{R_i R_{i+1}} + (D_{R_i R_{i+1}} + D_{R_i R_{i+1}})[i \bmod c = 0]); j + [\sum_q^{|R_j|} R R_q < d])$$

Таким чином ми отримуємо множину маршрутів, кожний елемент якої відповідає маршруту конкретного транспортного засобу, при чому всі маршрути мають довжину яка відрізняється одна від одної не більше ніж на відстань між двома максимально віддаленими точка на карті (в найгіршому випадку).

2.4.2 Дослідження властивостей задачі

Розглянемо завдання складання розкладу перевезень в умовах недовизначеності показників оцінювання для об'єкту транспортування.

Нехай ми маємо теоретико-множинну модель предметної області:

$$G = \langle V, Z, R, V', L, P, K', T \rangle$$

Де V – множина ВО. ВО можуть бути встановлені по декілька в одному місці;

Z – топологія розміщення ВО та місце доставки вантажу v'' , що для кожного i – го ВО $v_i \notin V$ встановлює географічні координати Z_i ;

C – множина характеристик, наявних безпосередніх шляхів сполучення з іншими ВО та місцем доставки вантажу, що в даній роботі узагальнюються показником вартості перевезення з пункту i в пункт j , $C = \{c_{ij}\}, i \in V, j \in V$);

V' – множина ВО, що повинна бути відвантажена протягом горизонту планування.

Для кожного $v_i \notin V'$ для можливості планування потрібна оцінка показника ваги $s.w_i = [w_{i\max}; w_{i\min}]$. У якості такої оцінки ми можемо використати лише мінімальну та максимальну статистичну оцінку кількості ВО обслугованих ТЗ певного типу за один рейс:

L – наявний парк ТЗ з параметрами цих засобів та кількістю одиниць кожного виду. Кожному транспортному засобу відповідає один водій;

P – тривалість робочої зміни водіїв ТЗ;

R – розклад обслуговування V' ВО на зміну;

K' – множина рейсів (кластерів), запланована на зміну для усіх ТЗ;

T – множина показників тривалості виконання змінного завдання кожною машиною(водієм): $t_l \in T, t_l = \sum_{k \in K', l \in L} t_{kl}$.

Потрібно виконати завдання складання розкладу перевезень на зміну так, щоб різниця у навантаженні на кожного з водіїв, відхилення реально перевезеного V'' від замовленого об'єму перевезень на зміну та вартість виконання всього об'єму

перевезень були мінімальними при умові не перевищення тривалості робочої зміни водіїв. Тобто:

$$\arg \min_{t_l, t_{l'} \in T} \max_{l, l' \in L} |t_l - t_{l'}|; \quad t_1 = f(R), \quad t_l \leq P, \quad \text{для } \forall l \in L$$

$$c = \sum_{i \in V'} \sum_{j \in V'} \sum_{l \in L_c} c_{ij} \rightarrow \min$$

$$(|V'| - |V''| = f'(R)) \rightarrow \min$$

Обмеження:

$$\sum_{i \in V'} \sum_l x_{ijl} \geq 1, j = 1, 2, \dots, |V'|$$

$$\sum_{i \in V'} x_{ijl} = \sum_{i \in V'} x_{jil}, j = 1, 2, \dots, |V'|, l = 1, 2, \dots, |L|$$

$$u_{ilk} - u_{jlk} + nx_{ijk} \leq |V'| - 1, j = 1, 2, \dots, |V'|, l = 1, 2, \dots, |L|, k = 1, 2, \dots, |K|$$

$$w_i \sum_{l \in L, i \in k_n, j \in k_n} x_{ijl} \leq W, n = 1, 2, \dots, |K|$$

Де $x_{ijl} = 1$ якщо ТЗ l їде напряму від ВО i до ВО j ; u_{ilk} – позиція ВО i у маршруті для ТЗ l у k -му рейсі.

Специфічні умови завдання: початковий пункт, місце призначення вантажу та кінцевий пункт прибуття однакові для всіх ТЗ. Наразі усі ТЗ мають однакову місткість та ВО мають однаковий тип. Один ВО може бути обслугованим лише одним ТЗ. Оскільки кожен рейс закінчується відвантаженням у одну точку призначення, то тільки ця точка може повторюватися у маршруті. Перевантаження ТЗ не припустимо і визначається. Матриця відстаней між ВО задовольняє нерівності трикутника.

Визначення: рейсом ТЗ будемо називати один цикл обслуговування ВО-в від моменту розвантаження/виїзду з гаража на початку зміни (початкова точка рейсу) до моменту відвантаження /заїзду в гараж в кінці зміни (кінцева точка рейсу).

2.4.3 Алгоритм розв'язання задачі

Для алгоритму складання розкладу перевезень необхідно 2 допоміжних алгоритми: вирішення задачі комівояжера та алгоритм кластеризації. Наведемо нижче їх реалізацію, а в подальших кроках будемо використовувати як функції TSP

та CLUSTER відповідно. Також одразу введемо умовні позначення системних функцій які необхідні для реалізації алгоритму:

- MIN – визначає мінімальне значення серед двох чисел;
- ROUTE_L – визначає довжину маршруту;
- LAST – повертає останній елемент масиву;
- ADD – додає новий елемент до масиву;
- SIZE – повертає розмір масиву;
- RANDOM_ROUTE – генерує випадковий маршрут;
- SWAP_CITIES – випадковим чином міняє елементи в маршруті місцями;
- RAND – повертає випадкове число в діапазоні від 0 до 1;
- EXP – функція експоненти;
- REVERT_SWAP – обернена дія до SWAP_CITIES;
- NOW – повертає поточний час;
- IS_EMPTY – повертає індикатор того пустий масив чи ні.

Псевдокод алгоритму кластеризації:

```

1  Вхід:          base // об'єкт який відповідає звалищю
2                      points // точки для яких проводиться кластеризація
3                      distances // матриця відстаней для заданих точок
4                      clustersAmount // кількість кластерів які необхідно згенерувати
5                      avgCarSize // середнє значення ємності автомобіля
6  Вихід:          clusters // сгенеровані кластери
7  Ініціалізація: accum = 0 // змінна в яку накопичується відстань
8                      clusters = [] // обраховані кластери
9                      tspPoints = TSP(points, distances, base) // обрахований маршрут
10 Розрахунок кластерів:
    for i:=1 to points.size do
11        toIndex = MIN(i + avgCarSize, points.size)
12        route = tspPoints[i, toIndex]
13        accum += ROUTE_L(route)
14  End
```

```

15   avgKmPerCar = accum / clustersAmount
16   for point in points do
17       currCluster = clusters.LAST
18       if ROUTE_L(curCluster) < avgKmPerCar or clusters.SIZE = clustersAmount
19           prevPoint = currCluster.LAST
20           If currCluster.SIZE % avgCarSize = 0
21               currCluster.ADD(base)
22           Else
23               currCluster.ADD(point)
24           Endif
25       Else
26           clusters.ADD(NEW_CLUSTER)
27       Endif
28   End

```

Діаграма послідовностей для алгоритму кластеризації наведена у додатку А на плакаті 6. Даний алгоритм отримує на вхід базовий об'єкт який відповідає звалищу, набір точок для яких необхідно провести кластеризацію, матрицю відстаней для заданих точок, а також середні показники ємності автомобіля. На вихід алгоритм повертає згенеровані кластери.

Псевдокод алгоритму імітації відпалу для вирішення задачі комівояжера:

```

1   Вхід:           points // точки для яких проводиться розрахунок маршруту
2                   distances // матриця відстаней для заданих точок
3                   numberOfIt // максимальна кількість ітерацій
4                   initialT // початкова температура відпалу
5                   resultT // кінцева температура відпалу
6   Вихід:          bestRouteL // довжина найкращого знайденого шляху
7                   bestRoute // найкращий знайдений шлях
8   Ініціалізація: bestRoute = RANDOM_ROUTE // на першій ітерації
                   найкращим приймається випадковий шлях

```

```

9           bestRouteL = ROUTE_L(bestRoute) // довжина поточного
найкращого шляху
10          currentRoute = bestRoute // прийняття найкращого шляху за
поточний
11          T = initialT // значення температури
12  Імітація відпалу:
    for i:=1 to numberOfIt do
13      currentRoute = SWAP_CITIES(currentRoute)
14      currentRouteL = ROUTE_L(currentRoute)
15      if currentRouteL < bestRouteL
16          bestRouteL = currentRouteL
17          bestRoute = currentRoute
18      Else
19          if RAND < EXP((bestRouteL-currentRouteL) / T)
20              bestRouteL = currentRouteL
21              bestRoute = currentRoute
22          Else
23              REVERT_SWAP
24          Endif
25      Endif
26      T = initialT / i
26      If T < resultT
27          Break
28      Endif
29  End

```

Діаграма послідовностей для алгоритму розв'язання задачі комівояжера наведена у додатку А на плакаті 7. Даний алгоритм отримує на вхід масив точок для яких необхідно вирішити задачу, матрицю відстаней, максимальну кількість ітерацій, а також початкове та кінцеве значення температури відпалу. На вихід алгоритм повертає найкращий шлях який йому вдалося згенерувати, а також його довжину.

Псевдокод алгоритму складання розкладу перевезень:

```

1  Вхід:          day //
2                      distances //
6  Вихід:         routes //
8  Ініціалізація: startTime = NOW
9                      clusters = CLUSTERS
10                     points = day.points
11                     realCarCapacity = day.realCarCapacity
12                     carIndex = 0
13 Розрахунок розкладу:
    for cluster in clusters do
14     loopTime = startTime
15     clusterRoute = TSP(cluster)
16     while not clusterRoute.IS_EMPTY
17         routeLength = ROUTE_LENGTH(clusterRoute)
18         toIndex = MIN(realCarCapacity, clusterRoute.SIZE)
19         routePoints = clusterRoute[0, toIndex]
20         willBeFreeAt = loopTime + (routeLength / CAR_SPEED)
21         loopTime = willBeFreeAt
22         routes.ADD(routePoints)
23     End
24 End

```

Діаграма послідовностей для алгоритму складання розкладу перевезень наведена у додатку А на плакаті 5. На вхід алгоритм отримує об'єкт з даними згенерованої зміни, а також матрицю відстаней для всіх об'єктів які входять в цю зміну. На вихід алгоритм повертає набір згенерованих маршрутів з яких буде складатися зміна.

2.5 Експериментальні результати досліджень

Проведемо експериментальні обрахунки з метою перевірки правильності складання математичної моделі. Обрахунки будемо проводити для трьох різних наборів вхідних даних кожен з яких буде відображати певну категорію налаштування алгоритму:

- мала місткість авто відносно кількості об'єктів які на нього виділені, тобто велика кількість рейсів. Для зручності введемо умовну назву – “min”;
- збалансована місткість авто відносно кількості об'єктів які для нього виділені. Для зручності введемо умовну назву – “avg”;
- велика місткість авто відносно кількості об'єктів які на нього виділені, тобто мала кількість рейсів. Для зручності введемо умовну назву – “max”.

Всі три експеримент будемо проводити для кожного з методів розв'язання задачі планування розкладу перевезень, а саме для:

- класичного підходу. Умовна назва – “classical”;
- методу динамічного перерозподілу. Умовна назва – “dynamic”;
- методу рівномірно розподілу роботи на етапі планування. Умовна назва – “uniform”.

Наведемо вхідні дані для експериментів “min”, “avg”, “max” у таблицях 2.1, 2.2 та 2.3 відповідно.

Таблиця 2.1 – значення параметрів для експерименту “min”

Назва параметру	Величина
Кількість об'єктів	400
Кількість автомобілів	10
Ємність автомобіля	10
Похибка ємності	0.2

Таблиця 2.2 – значення параметрів для експерименту “avg”

Назва параметру	Величина
Кількість об’єктів	400
Кількість автомобілів	20
Ємність автомобіля	10
Похибка ємності	0.2

Таблиця 2.3 – значення параметрів для експерименту “max”

Назва параметру	Величина
Кількість об’єктів	400
Кількість автомобілів	40
Ємність автомобіля	10
Похибка ємності	0.2

Всі обрахунки будемо проводити для одного й того самого набору об’єктів на карті який складається з 400 точок, та однієї додаткової яка є базовою точкою початку та завершення всіх маршрутів, тобто звалищем.

Проведемо обрахунки для кожної з реалізацій алгоритму:

– “classical”:

1) “min”

Час виконання (години)	3138
Відстань (км)	860195
Кількість рейсів	50
Відстань подолана бригадою яка закінчила першою (км)	57006
Час бригади яка закінчила першою (години)	1425
Відстань подолана бригадою яка закінчила останньою (км)	125548
Час бригади яка закінчила останньою (години)	3138

2) “avg”

Час виконання (години)	2028
Відстань (км)	558592
Кількість рейсів	30
Відстань подолана бригадою яка закінчила першою (км)	38060
Час бригади яка закінчила першою (години)	951
Відстань подолана бригадою яка закінчила останньою (км)	81126
Час бригади яка закінчила останньою (години)	2028

3) “max”

Час виконання (години)	1485
Відстань (км)	399481
Кількість рейсів	20
Відстань подолана бригадою яка закінчила першою (км)	28089
Час бригади яка закінчила першою (години)	702
Відстань подолана бригадою яка закінчила останньою (км)	59439
Час бригади яка закінчила останньою (години)	1485

– “dynamic”:

1) “min”

Час виконання (години)	2808
Відстань (км)	989489
Кількість рейсів	53
Відстань подолана бригадою яка закінчила першою (км)	86074
Час бригади яка закінчила першою (години)	2151
Відстань подолана бригадою яка закінчила останньою (км)	112340
Час бригади яка закінчила останньою (години)	2808

2) “avg”

Час виконання (години)	1821
Відстань (км)	604149
Кількість рейсів	28
Відстань подолана бригадою яка закінчила першою (км)	42521
Час бригади яка закінчила першою (години)	1063
Відстань подолана бригадою яка закінчила останньою (км)	72856
Час бригади яка закінчила останньою (години)	1821

3) “max”

Час виконання (години)	1282
Відстань (км)	393281
Кількість рейсів	17
Відстань подолана бригадою яка закінчила першою (км)	29785
Час бригади яка закінчила першою (години)	744
Відстань подолана бригадою яка закінчила останньою (км)	51304
Час бригади яка закінчила останньою (години)	1282

– “uniform”:

1) “min”

Час виконання (години)	2430
Відстань (км)	872974
Кількість рейсів	51
Відстань подолана бригадою яка закінчила першою (км)	79025
Час бригади яка закінчила першою (години)	1975
Відстань подолана бригадою яка закінчила останньою (км)	97205
Час бригади яка закінчила останньою (години)	2430

2) “avg”

Час виконання (години)	1409
Відстань (км)	490742
Кількість рейсів	26
Відстань подолана бригадою яка закінчила першою (км)	39560
Час бригади яка закінчила першою (години)	989
Відстань подолана бригадою яка закінчила останньою (км)	56367
Час бригади яка закінчила останньою (години)	1409

3) “max”

Час виконання (години)	1139
Відстань (км)	332653
Кількість рейсів	15
Відстань подолана бригадою яка закінчила першою (км)	21820
Час бригади яка закінчила першою (години)	545
Відстань подолана бригадою яка закінчила останньою (км)	45564
Час бригади яка закінчила останньою (години)	1139

Як можна побачити з результатів проведення експериментів, всі з реалізацій дійсно показали себе саме так як від них і очікували, а саме:

- класичний підхід – має найкращу відстань, але середні показники часу виконання;
- метод динамічного перерозподілу – має найгіршу відстань та середні показники часу виконання;
- метод рівномірно розподілу роботи на етапі планування – має близькі до найкращих показники відстані та однозначно найкращі показники часу виконання.

Графіки з результатами експериментів наведено у додатку А на плакаті 3. Як можна побачити з отриманих результатів, метод рівномірного розподілу роботи на етапі планування показує показники за часом на 19% кращі ніж два інших алгоритми,

при цьому розбіжність у показниках відстані з класичним методом настільки незначна, що її можна віднести до статистичної помилки.

Висновок до розділу

В цьому розділі було сформовано математичну модель для вирішення задачі складання розкладу перевезень в умовах недовизначеності показників.

В розділі викладена математична формалізація подібного процесу, проаналізовані різні критерії та методи оптимізації а також сформовані сильні та слабкі сторони кожного із них.

Проведені розрахунки для кожного з трьох алгоритмів оптимізації для трьох основних категорій конфігурацій маршруту, на підставі чого було виявлено основні підходи які необхідно використати для опису фінального варіанту алгоритму.

На підставі знайдених залежностей був описаний алгоритм який дозволяє оптимізувати показник мінімізації розриву в часі виконання для кожної із обслуговуючих бригад, при цьому зберігаючи показники відстані на тому самому рівні що і класична реалізація, а також враховує недовизначеність показників починаючи з самого першого етапу планування маршруту.

3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Засоби розробки

Створення програмного забезпечення було вирішено проводити з використання наступних мов програмування:

- Java – для реалізації серверної частини;
- JavaScript – для реалізації клієнтської частини.

Мови програмування

а) Java – на даний момент ця мова є найбільш популярним інструментом для виконання розробки серверної частини, для цього є декілька причин:

- 1) це повністю об'єктно-орієнтована мова програмування;
- 2) мова забезпечує автоматичне збирання сміття та просте управління пам'яттю, крім того, має функціонал узагальнених типів даних [37]. Все це робить її дуже зручною для написання серверної частини;
- 3) статична перевірка типів даних під час компіляції та перевірки коректності під час запуску робить її дуже надійною мовою;
- 4) компіляція в байт-код [38] дозволяють віртуальній машині Java (JVM) [39, 40, 41] бути досить швидкою, через що, Java має високу продуктивність;
- 5) Java дозволяє багатопоточне виконання [42];
- 6) Java додаток можна запустити на будь-якій системі з встановленою JVM. Ця портативність є головною перевагою;
- 7) масштабованість – це дуже важлива перевага яка досягається шляхом того, що екземпляри JVM можуть бути дуже просто клоновані та додані в потрібній кількості для забезпечення потрібної пропускну здатності.

б) JavaScript – даний момент це єдина мова яка підтримується всіма браузерами та відповідає за динамічну складову вигляду сторінки, серед основних її переваг можна відмітити:

- 1) інтерактивність робить JavaScript дуже потужним інструментом, який дає можливість зробити багато корисних речей для веб-сайту, а саме JavaScript дозволяє створювати динамічні компоненти інтерфейсу. Це

надає сайту додаткову функціональність, яку в іншому випадку неможливо досягти лише за допомогою HTML та CSS. JavaScript дозволяє веб-сторінкам реагувати на активність користувачів і динамічно оновлювати себе, і все це, без необхідності оновлення сторінки;

- 2) універсальність – наразі існує велика кількість веб-браузерів які в свою чергу можуть працювати на різних операційних системах та пристроях. Саме тому обираючи технологію розробки необхідно обирати технології які б однаково підтримувалися та працювали на всіх комбінаціях браузерів та операційних систем. JavaScript – чудово справляється з цією задачею завдяки строгій стандартизації;
- 3) фреймворки – на даний момент JavaScript одна з найбільш стрімко розвиваючихся мов програмування, для якої постійно створюються все нові і нові фреймворки які дозволяють вирішувати нові задачі або спрощувати рішення для вже існуючих.

3.2 Вимоги до технічної частини

Система складається з двох фізичних серверів, один з яких використовується для розгортання клієнтської частини а другий для серверної.

Сервер

Програмне забезпечення не має жодних конкретних вимог та обмежень щодо операційної системи та технічного забезпечення серверу, це досягається завдяки кросплатформенній властивості джави. Це означає, що в ролі серверу може бути використаний комп'ютер як під управлінням операційної системи Linux [43] так і Windows [44]. При необхідності дуже просто можна додати конфігурацію для загорання додатку в Docker контейнер [45, 46], що забезпечить ще більшу сумісність та портативність.

Для коректної роботи серверної частини необхідно задовільнити мінімальні технічні умови для серверу на якому додаток буде розгорнуто, а саме:

- процесор з тактовою частотою не нижче за 1.5 ГГц;
- об'єм оперативної пам'яті не менше 8 Гб;

- HDD [47] або SSD [48] об'ємом не менше 50 Гб.

Додатково має бути встановлене таке програмне забезпечення:

- база даних PostgreSQL 9.0 або вище;
- на вибір JRE [49] версії 1.8 або вище або Docker.

Клієнт

Сервер для клієнтської частини так само як і для серверна частина, не залежить від операційної системи завдяки можливості докеризування системи, а також через те, що весь необхідний софт існує як для Linux так і для Windows. Для клієнтської частини вимоги до потужності серверу будуть дещо нижчими, так як всі основні обрахунки виконуються на серверній частині:

- процесор з тактовою частотою не нижче за 1.2 ГГц;
- об'єм оперативної пам'яті не менше 4 Гб;
- HDD або SSD об'ємом не менше 30 Гб.

Додатково має бути встановлене таке програмне забезпечення:

- база даних nginx;
- на вибір NodeJs [50] версії 10.0 або вище або Docker.

3.3 Вимоги до програмного продукту

Розглянемо окремо функціональні вимоги до клієнтської та серверної частин.

Функціональні вимоги до серверної частини програмного забезпечення програмно-інформаційної системи. Система повинна:

- обробляти одночасно запити від великої кількості користувачів;
- забезпечувати можливість створення аккаунту та входу в систему;
- надавати коректні результати планування розкладу перевезень;
- формувати статистику по виконаній роботі;
- виступати в ролі CRM-системи для транспортних компаній які займаються перевезеннями сміття;
- реагувати на запити від клієнтської системи які надходять по протоколу HTTP [51] або HTTPS [52] у форматі JSON [53];

- коректно реагувати на можливі помилки та інформувати про це в інформативному для клієнтів форматі.

Функціональні вимоги до клієнтської частини програмного забезпечення програмно-інформаційної системи. Клієнтська частина повинна:

- відображати інформацію щодо запланованих перевезень;
- надавати форму для формування запитів на складання розкладу перевезень;
- відображати статистику по виконаній роботі;
- відображати заплановані маршрути на карті;
- відображення отриманих з серверу об'єктів на карті;
- відправляти запити на сервер по протоколу HTTP або HTTPS у форматі JSON;
- коректно відображати користувачу інформацію у випадку внутрішніх помилок серверної частини.

3.4 Архітектура програмного забезпечення

Архітектура системи виконана за клієнт-серверною технологією. Подібне рішення було обрано в зв'язку з тим, що необхідно мати єдиний, централізований вузол керування системою, в ролі якого саме і виступає сервер.

Сервер також виступає в ролі планувальника задач, та саме не ньому відбувається розрахунок розкладу перевезень та вирішення дуже ресурсозатратних задач знаходження найкоротшого шляху. Саме завдяки такому підходу сервер виступає в ролі координатора та контролює загальний стан виконання зміни та розподіляє роботу між виконавцями.

При використанні такої архітектури ініціатором початку взаємодії є клієнтська частина, оскільки сервер не має можливості знати про всіх своїх клієнтів.

Сервер може виконувати планування розкладу в двох режимах:

- за запитом адміністратора – адміністратор заходить на веб-сайт додатку, переходить в особистий кабінет, обирає бажані параметри які мають бути використані при складанні розкладу та нажимає на кнопку “Розрахувати”
- за розкладом заданим в Cron Job [54] – адміністратор може задати в Cron форматі розклад згідно якому розклад буде формуватися автоматично.

3.4.1 Клієнтська частина

Клієнтська частина реалізована завдяки фреймворку VueJS [55]. Цей фреймворк є надзвичайно зручним при створенні малих та середніх за розміром клієнтських застосунків, також він признаний відносно простим та гарним рішенням для того, щоб почати своє знайомство з фронтендом саме з нього – як і було в моєму випадку.

Зважаючи на те, що всі обрахунки та збереження даних відбувається на серверній частині, клієнтська не має потреби у використанні жодних сховищ даних, та отримує всі необхідні для функціонування дані за допомогою HTTP у форматі JSON з серверної частини. Нижче наведена структура формату основних запитів та відповідей на них які використовуються для відображення карти та маршрутів, а також відповідної їм статистики. Для зручності та лаконічності приклади запитів на генерацію заявки на створення розкладу та розрахунок розкладу перевезень наведемо у форматі CURL[56] у таблиці 3.1.

Таблиця 3.1 – Структура запитів до серверу

Назва запиту	Request	Response
Генерація заявки на створення розкладу перевезень	<pre>curl -X POST \{host}/day \ -H 'Accept: */*' \ -H 'Content-Type: app/json' \ -d '{ "pointsAmount": 50, "carsAmount": 5, "carCapacity": 5, "carCapacityError": 0.2 }'</pre>	<pre>{ "dayId": 12, "maxCoordinate": 47.95, "routes": [{ "points": [{ "lng": -7.81, "lat": -3.41}, ...], "shell": [{ "lng": -7.81, "lat": -3.41 }, ...] }, ...], "convexP": [{ "lng":-48.8, "lat": -22.56 },...], "allPoints": [{ "lng":-48.8, "lat": -22.56 },...], "base": { "lng": -60.0, "lat": 35.0,"id": 1,"base": true } }</pre>

Назва запиту	Request	Response
Розрахунок розкладу перевезень	<pre>curl -X POST \ {host}/day/simulate/12 \ -H 'Accept: */*' \ -H 'Accept-Encoding: gzip' \ -H 'Cache-Control: no-cache' \ -H 'Connection: keep-alive' \ -H 'Content-Length: 0' \ -H 'Content-Type: app/json' \ -H 'Host: localhost:8080</pre>	<pre>{ "id": 12, "date": "2019-11-02", "completed": true, "startTime": "2019-11-02T13:13:15.6", "endTime": "2019-12-27T00:16:06", "totalDistanceKm": 231090.21, "carCapacityError": 0.2, "carCapacity": 5, "cars": [{ "id": 1 }, ...], "points": [{ "lng": -48.8, "lat": -22.56 }, ...], "routes": [{ "id": 341, "startTime": "2019-11-02T13:13:15", "endTime": "2019-11-18T01:46:51", "distance": 14902.404, "points": [{ "lng": -14.75, "lat": 23.41 }, ...], "convexHull": [{ "lng": -14.75, "lat": 23.41 }, ...], "car": { "id": 1 } }, ...] }</pre>

Запит на генерацію заявки на створення розкладу перевезень містить поля які наведено в таблиці 3.2.

Таблиця 3.2 – Значення полів запиту на генерацію заявки

Назва поля	Призначення
pointsAmount	Кількість точок які будуть приймати участь в розрахунку
carsAmount	Кількість транспортних засобів які виділяються на зміну
carCapacity	Ємність кожного транспортного засобу
carCapacityError	Відсоток похибки на який може відрізнятись ємність транспортного засобу

Пояснення значень відповіді на цей запит наведено в таблиці 3.3.

Таблиця 3.3 – Значення полів відповіді на генерацію заявки

Назва поля	Призначення
dayId	Унікальний ідентифікатор згенерованої зміни
routes	Масив попередньо розрахованих кластерів
points	Масив точок які входять в кластери
lng, lat	Географічні координати кожної з точок
shell	Масив точок які складають мінімальну опуклу оболонку[49] яка необхідна для графічного відображення кластеру
lng, lat	Географічні координати кожної з точок
convexPolygon	Масив точок які складають мінімальну опуклу оболонку для об'єктів всієї зміни
lng, lat	Географічні координати кожної з точок
allPoints	Масив всі точок які входять в зміну
lng, lat	Географічні координати кожної з точок
base	Базова точка (звалище)
lng, lat	Географічні координати кожної з точок
id	Ідентифікатор точки звалища
base	Ідентифікатор того, чи є об'єкт базовим чи ні
maxCoordinate	Значення максимальної географічної координати необхідне для графічного відображення кластеру

Запит на розрахунок розкладу перевезень не містить жодної інформації в тілі HTTP запиту, тому в таблиці 3.4 наведемо пояснення лише для відповіді яка приходить на цей запит.

Таблиця 3.4 – Значення полів відповіді на запит розрахунку розкладу

Назва поля	Призначення
id	Ідентифікатор складеного розкладу
date	Дата опрацювання розкладу
completed	Стан того чи є розклад опрацьованим
startTime	Час початку опрацювання розкладу
endTime	Час завершення опрацювання розкладу
totalDistanceKm	Загальна відстань яку пройшли всі ТЗ за зміну
carCapacityError	Відсоток похибки місткості ТЗ
carCapacity	Місткість ТЗ
cars	Масив з інформацією про ТЗ які приймають участь у зміні
id	Ідентифікатор ТЗ
points	Масив з точками які приймають участь у зміні
lng, lat	Географічні координати кожної з точок
routes	Масив з усіма маршрутами зміни
startTime	Час початку маршруту
endTime	Час завершення маршруту
distance	Довжина маршруту
points	Точки з яких складається маршрут
lng, lat	Географічні координати кожної з точок
convexHull	Масив з точками мінімальної опуклої оболонки маршруту
lng, lat	Географічні координати кожної з точок
car	ТЗ який опрацював маршрут
id	Ідентифікатор ТЗ

3.4.2 Серверна частина

Архітектура серверної частини виконана за мікровервісним принципом, такий підхід є більш сучасним та дозволить більш гнучко регулювати кількість обчислювальної потужності [58] необхідний для кожної складової, що в свою чергу позитивно відобразиться на раціональності витрат на оренду серверного обладнання та дозволить мінімізувати вартість проекту.

Кожний мікросервіс можна розглядати як окреме застосування яке повністю обслуговує свою частину функціоналу незалежно від інших сервісів.

Крім того такий підхід забезпечить значно легше масштабування у випадку підвищення попиту на систему.

Архітектура

Структура мікросервісів представлена в додатку А на плакаті 8. На ньому ми також можемо побачити які компоненти з якими спілкуються. З плакату видно що користувач має доступ лише до клієнтського сервісу для якого відкритий доступ до 3х сервісів серверної частини. Безпосередньо до бази даних доступ відкритий лише для трьох серверних мікросервісів.

Подібна структура з обмеження доступу є гарною практикою з точки зору безпеки та надійності, оскільки мінімізується потенційна кількість можливих способів атаки та отримання доступу до бази даних. Діаграма побудови архітектури наведена в додатку А на плакаті 8.

Розглянемо кожний мікросервіс окремо за його функціоналом та виділимо його роль в системі:

- **auth** – даний мікросервіс відповідає за авторизацію в системі. Він єдиний до якого клієнтський сервіс може звертатися без авторизаційного токена. Працює він наступним чином – коли користувач бажає отримати доступ до функціоналу який передбачає авторизований доступ, він надсилає свої дані доступу та як результат отримує токен який має життєвий цикл обмежений за часом. Після цього клієнт зберігає собі цей токен та додає його до кожного запиту який він робить до двох інших сервісів. Авторизаційний сервіс в свою чергу кожний раз перевіряє чи не вичерпався строк придатності у токена і

якщо таке трапляється то доступ обмежується до того часу поки не буде отримано новий токен;

- **platform** – це головний сервіс який відповідає за основну бізнес-логіку системи, саме завдяки ньому відбувається координування всієї системи. Він надає API для додавання нових водіїв та об'єктів, відображення статистики та формування розкладу у вигляді який буде зручний для клієнтського сервісу. Також в цьому сервісі відбувається вся логіка обробки помилок та створення Cron процесів які можуть автоматично запускати процеси складання розкладу перевезень;
- **algorithm** – вся логіка обрахунку алгоритмів складання розкладу перевезень та розрахунку маршрутів винесена в цей сервіс. Зважаючи на те, що дані процеси потребуються максимальну кількість обчислювальної потужності цей сервіс є найбільш дорогим з точки зору плати за оренду ресурсів хмарних провайдерів.

3.5 Інструкція користувача

Для початку роботи з системою необхідно:

- пройти форму авторизації, яка представлена на рисунку 3.1;

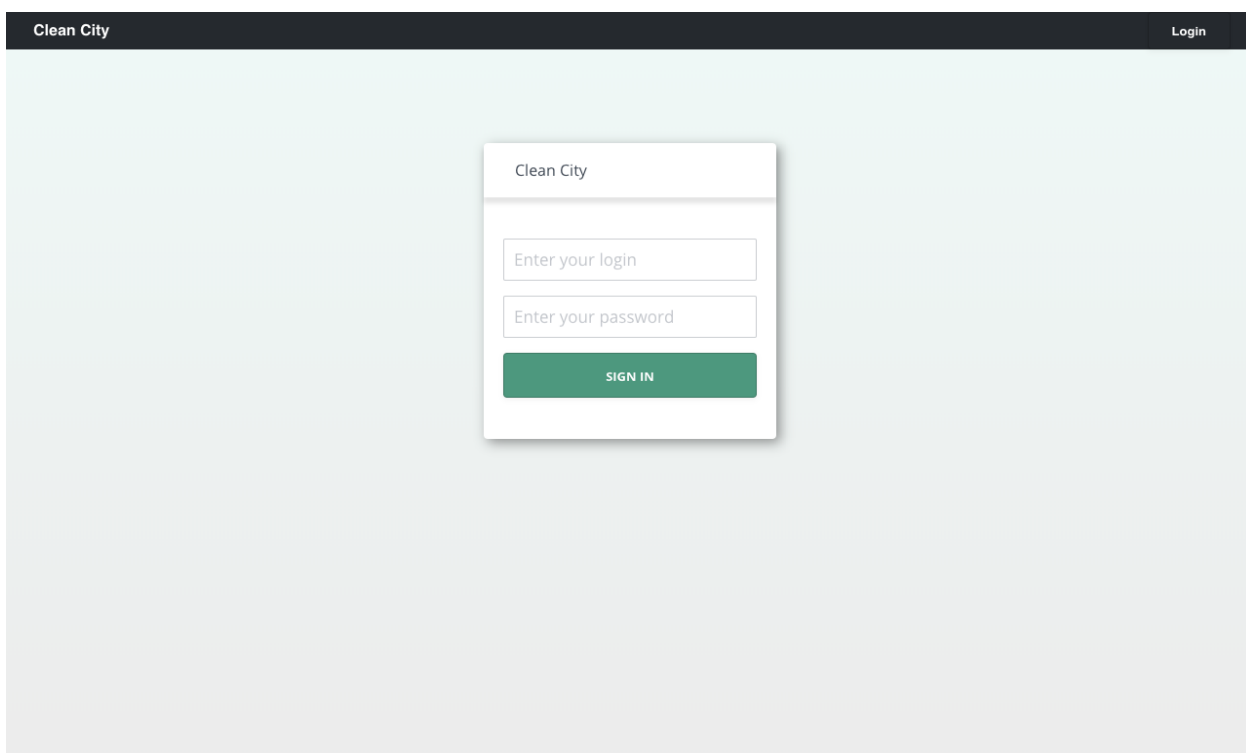


Рисунок 3.1 – форма авторизації

– після цього користувач потрапляє на сторінку на якій відображаються всі об'єкти обслуговування які закріплені за його акаунтом – рисунок 3.2. Також для зручності користування на даній сторінці відображається мінімальна оболонка яка оточує всі об'єкти;

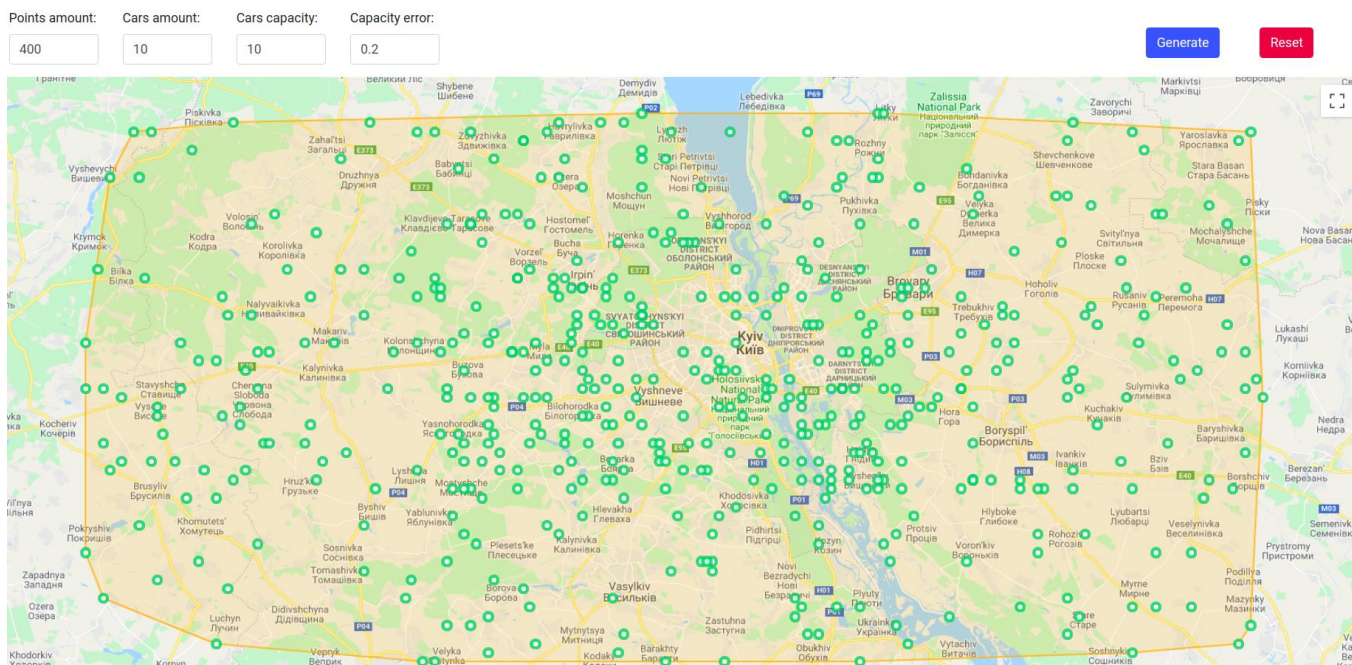


Рисунок 3.2 – Відображення всіх об'єктів які відносяться до акаунту користувача

– на цьому етапі користувач має можливість задати параметри для генерації розкладу такі як:

- 1) кількість об'єктів для яких необхідно скласти розклад;
- 2) кількість автомобілів які приймають участь у зміні;
- 3) місткість автомобіля;
- 4) відсоток похибки на яку може відрізнятись ємність автомобіля.

– після того як всі параметри задані, користувач має можливість згенерувати зміну, результат показано на рисунку 3.3.

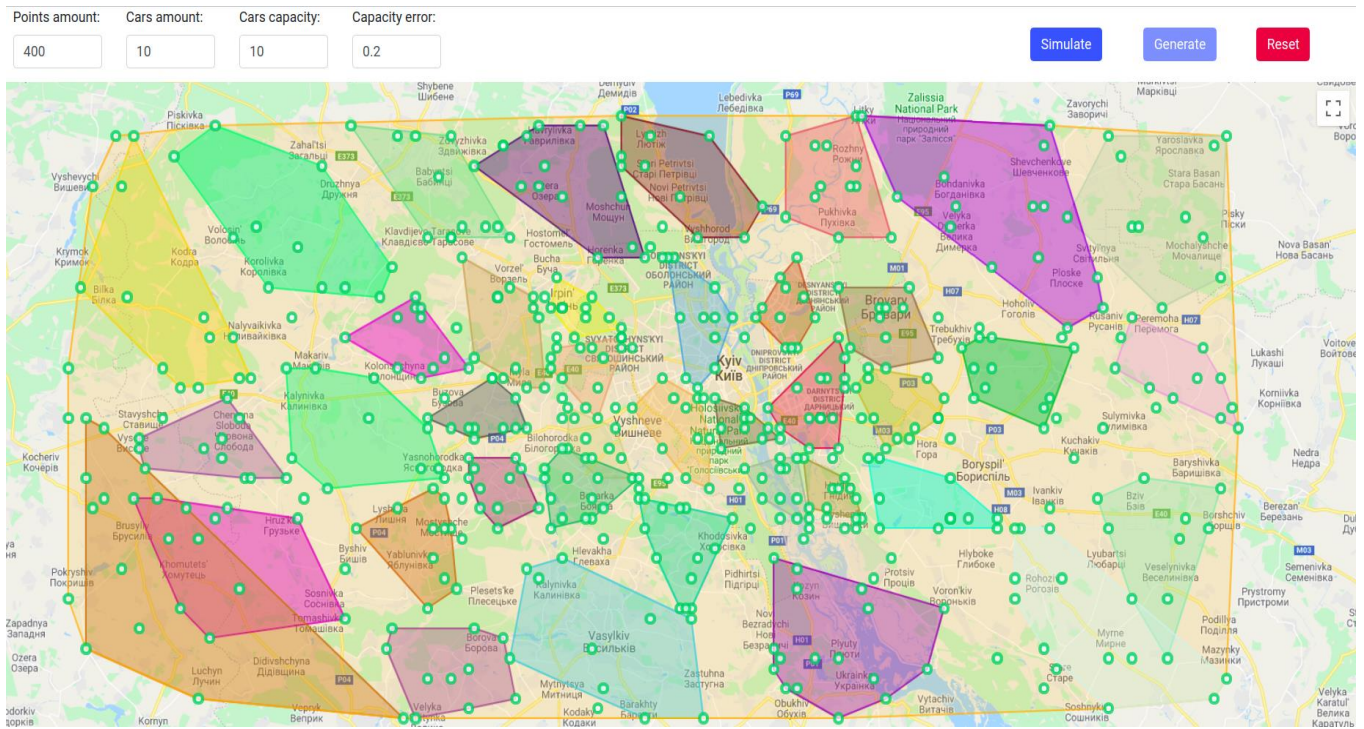


Рисунок 3.3 – Відображення початкового розподілу кластерів

- на даному етапі користувач має змогу ознайомитися з початковою схемою розподілення кластерів. Якщо користувача задовольняє той попередній результат який він бачить, користувач може далі натиснути кнопку “simulate” для того, щоб безпосередньо розпочати планування розкладу перевезень, або у випадку, якщо користувач змінив свою думку, він може натиснути кнопку “reset” для того, щоб повернутися на попередній крок;

— на цьому етапі користувач має змогу переглядати згенеровані маршрути. Для того, щоб не перевантажувати інтерфейс, всі маршрути відображаються без врахування звалища, тому насправді кожен з них містить ще одну точку яка не відображається. На цьому етапі користувач отримує можливість переглянути детальну статистику за допомогою кнопки “Info”, або повернутися на попередній крок за допомогою кнопки “Reset”. Те як це виглядає можна побачити на рисунку 3.4;

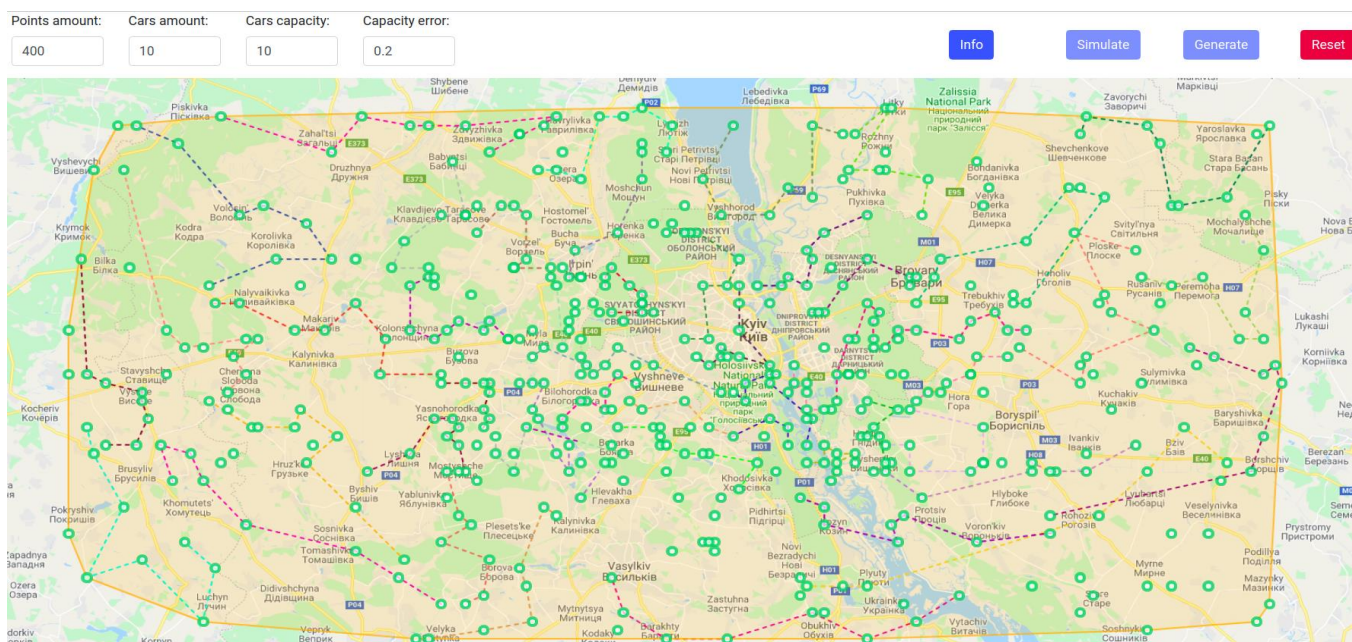


Рисунок 3.4 – Відображення згенерованих маршрутів

– після натиснення на кнопку “Info” користувачу стає доступним екран з детальною статистикою відносно всіх маршрутів, автомобілів, а також агреговані дані стосовно всієї зміни (рисунок 3.5). При повторному натисненні ця інформація зникає, проте може бути викликана знову. Далі єдиною опцією користувача є натиснення на кнопку “Reset” та повернення до початкового стану.

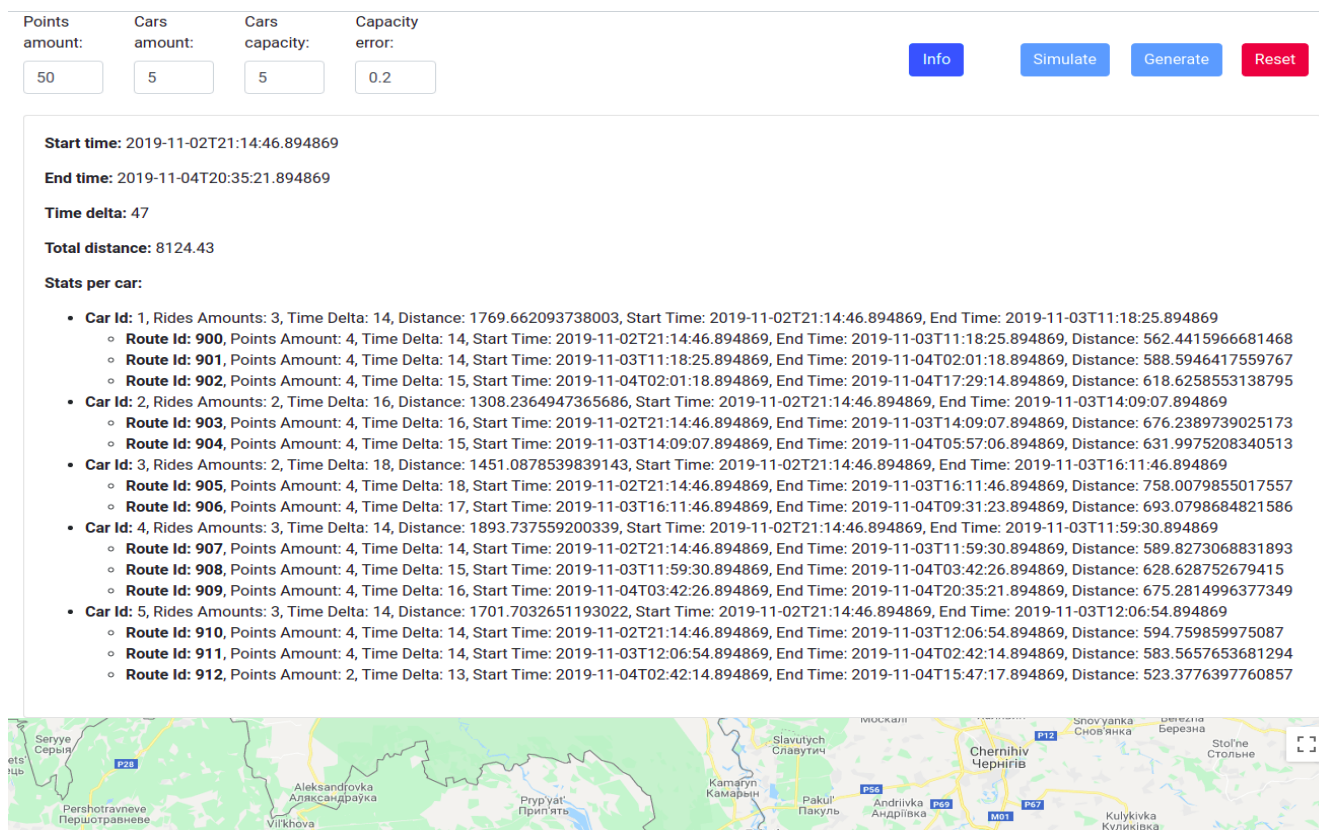


Рисунок 3.5 – Відображення детальної статистики

Висновок до розділу

Цей розділ було присвячено рішенням з програмного та технічного забезпечення. Задача як поставлена перед системою яку необхідно розробити – це складання розкладу перевезень та вирішення транспортних задач великої розмірності. Для досягнення цих цілей було обрано тип архітектури програмного забезпечення, а саме – архітектура клієнт-серверного типу.

Було проаналізовано та обрано список технологій які необхідно використати для клієнтської та серверної частин, а саме – клієнтська частина виконана за допомогою мови JavaScript та з використанням фреймворку VueJS, а серверна частина виконана за допомогою мови програмування Java та з використанням фреймворку Spring. Було детально розглянуто плюси і мінуси кожної з цих технологій, а також наведені обґрунтування того, чому саме ці технології були обрані для реалізації проекту.

Серверна частина виконана з використанням мікросервісного підходу, саме тому було описано всі наявні мікросервіси, та те як вони взаємодіють між собою. Було проведено аналіз на доцільність використання подібного підходу, та наведено всі переваги які він надає в розрізі запропонованої системи.

Було розроблено розділ з інструкціями користувача, який детально пояснює всі кроки взаємодії з системою, та допомагає розібратися в наявному функціоналі та тому як ним правильно користуватися.

4 РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ

4.1 Опис ідеї проекту

В межах даного проекту пропонується розробити систему для складання розкладу перевезень для спеціалізованих комунальних підприємств які займаються вивезенням сміття. Система в майбутньому може бути адаптована для використання в більш широкій предметній області, а саме для будь-яких компаній які мають потребу в складанні розкладу перевезень.

Відмінною рисою системи є те, що алгоритм складання розкладу перевезень направлений на оптимізацію кількості роботи яка виділяється на кожного водія транспортного засобу та мінімізує різницю в довжині робочої зміни для кожного з них.

Система спроектована за клієнт серверною технологією, завдяки чому вона може розповсюджуватися як “сервіс”, що значно спростить процес інтеграції для майбутніх клієнтів. Система може працювати як з реальними даними так і в режимі моделювання, що дозволить клієнтам більш наглядно оцінити переваги від її впровадження.

Призначенням цієї системи є створення продукту який комплексно вирішує задачу складання розкладу перевезень для транспортних компаній з великою кількістю автомобілів та об'єктів які опрацьовуються.

Сутність розробки – це створення системи яка б покращила загальні показники часу виконання робочої зміни без погіршення показників відстані.

Цільовою аудиторією є спеціалізовані комунальні підприємства які займаються вивезенням сміття.

Основними вигодами є зниження загального часу виконання зміни, що в свою чергу дорівнює мінімізації витрат та збільшенню кількості прибутків, також це позитивно відобразиться на рівні задоволеності працівників які будуть працювати в більш справедливих умовах праці.

4.2 Опис конкурентів

На етапі аналізу ринку, було виявленого головного конкурента який надає подібні послуги на українському ринку, це компанія Ant-Logistics. Ця компанія займає дуже впевнену позицію на ринку рішень для логістичних перевезень та має дуже гнучкий та потужний функціонал.

Вони виступають в ролі системи управління відносин з клієнтами, далі CRM [59], та надають компаніям-клієнтам можливість повністю вести логістичну складову свого бізнесу в межах їх системи.

Функціонал дозволяє:

- гнучко конфігурувати параметри автомобілів;
- вносити в систему інформацію про склади;
- вносити інформацію про місця доставки;
- задавати розклад роботи зміни як для водіїв так і для пунктів доставки та складів;
- коригувати створені маршрути в ручному режимі;
- переглядати історію та звіти перевезень.

Також їх продукт має й інший додатковий функціонал, але він не пересікається з предметною областю, в межах якої ми плануємо конкурувати.

Зовнішній вигляд веб-сайту системи-конкурента наведено на рисунку 4.1. На даному рисунку показано перегляд створеного маршруту та відображено вікно опцій для його редагування.

На рисунку 4.2 наведено вигляд декількох сторінок їх мобільного додатку, який був нещодавно представлений. Тут ми можемо побачити процес авторизації та перегляду вже створених маршрутів.

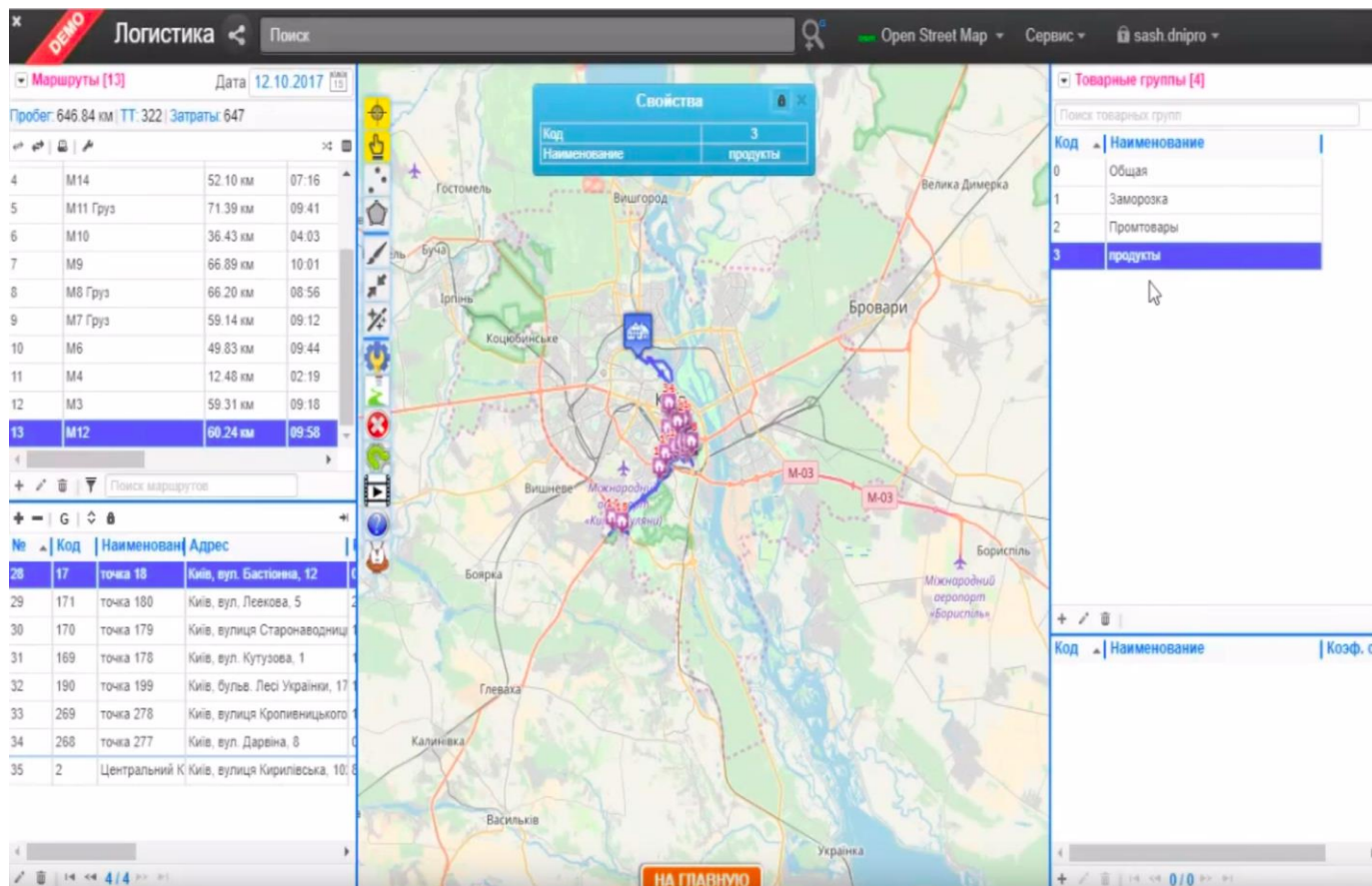


Рисунок 4.1 – Интерфейс веб-сайту додатку Ant Logistics

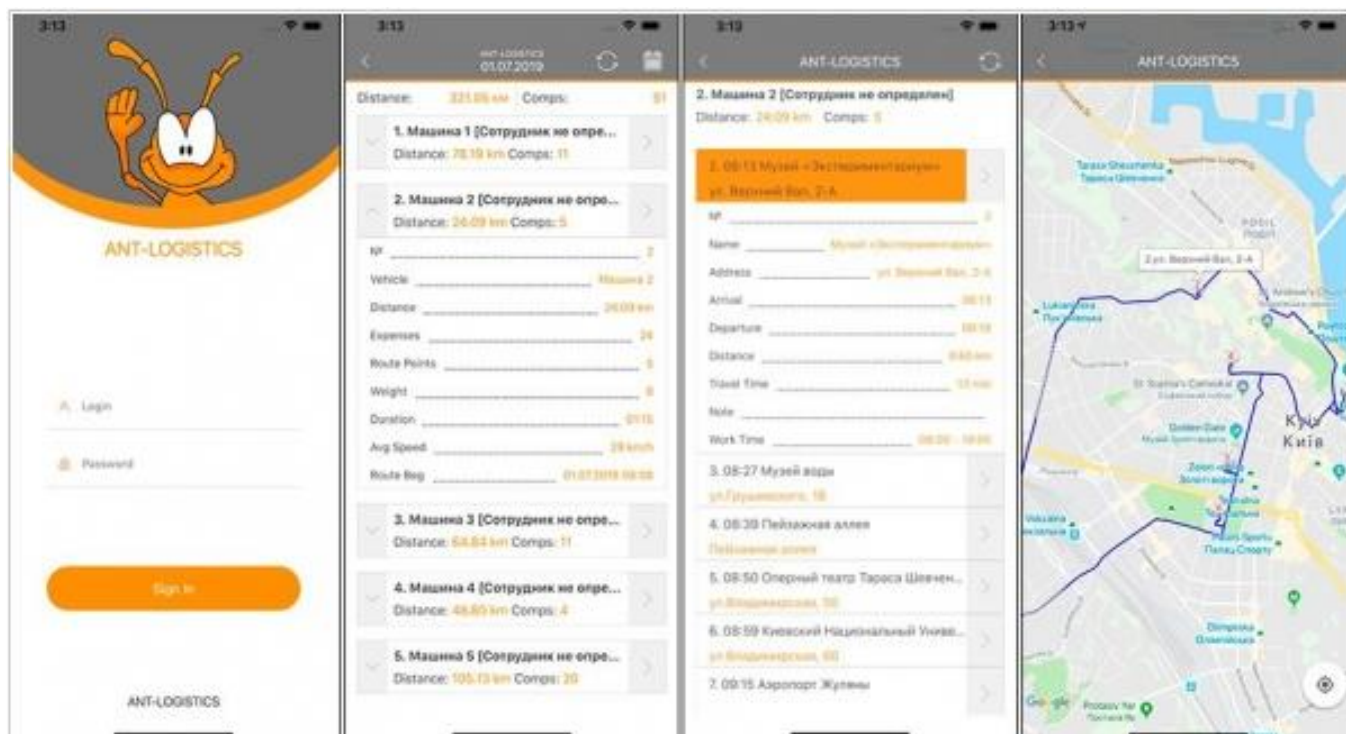


Рисунок 4.2 – Интерфейс мобильного додатку Ant Logistics

Спробуємо співставити між собою дві системи, та виділити слабкі та сильні сторони кожної з них. Ці дані згрупуємо та представимо у вигляді таблиці 4.1.

Таблиця 4.1 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

Ідея	(потенційні) товари/концепції конкурентів		Слабка сторона (W)	Нейтральна сторона (N)	Сильна сторона (S)
	<i>Запропонований стартап</i>	<i>Ant-Logistics</i>			
Складання маршрутів одразу для багатьох автомобілів	Функціонал доступний	Функціонал доступний		+	
Оптимізація за часом опрацювання	Функціонал доступний	Відсутній			+
Можливість інтеграції для клієнтів	Система має відкритий REST API [60, 61]	Система має відкритий REST API		+	
Трекінг даних стосовно місцезнаходження автомобілів	Функціонал доступний	Система пропонує інтегрувати цю функцію зі сторонніх сервісів			+
Формування звітів щодо виконаної роботи	Функціонал доступний	Функціонал доступний		+	

Базуючись на проведеному аналізі сильних, слабких та нейтральних характеристик обох систем можна зробити висновок про загальну конкурентоспроможність стартапу. Як ми можемо побачити, моя система має певні переваги які можуть бути цікавими для майбутніх клієнтів.

4.3 Технологічний аудит ідеї проекту

В таблиці 4.2 приведемо список технологій та обладнання яке необхідно використати для реалізації ідеї, на основі чого зробимо висновок про технічну здійсненність проекту.

Таблиця 4.2 – Технологічна здійсненність ідеї проекту

Ідея проекту	Технології реалізації	Наявність технології	Доступність технології
<i>Трекінг даних стосовно місцезнаходження автомобілів</i>	Мобільний додаток який буде виконувати полінг [62, 63] серверу та оновлювати координати	Наявна	Доступна
<i>Складання маршрутів одразу для багатьох автомобілів</i>	Spring Boot [64, 65]	Наявна	Доступна
<i>Можливість інтеграції для клієнтів</i>	REST API	Наявна	Доступна
<i>Оптимізація за часом опрацювання</i>	Алгоритм виконаний мовою Java	Наявна	Доступна
<i>Формування звітів щодо виконаної роботи</i>	Java Бібліотеки для формування PDF [66] та XLS [67] файлів	Наявна	Доступна

Переглянувши повний список необхідних технологій можна зробити висновок, що задумка проекту з технологічної точки зору здійсненна та може бути виконана без жодних обмежень.

4.4 Аналіз ринкових можливостей запуску стартап-проекту

Розглянемо питання, які можуть виникнути при впровадженні стартап-проекту на ринок. Щоб проаналізувати попит на проект порівнюємо головних конкурентів у таблиці 4.3.

Таблиця 4.3 – Попередня характеристика потенційного ринку стартап-проекту

Показник стану ринку	Характеристика
<i>Кількість визначених конкурентів</i>	Один – Ant-Logistics
<i>Розмір ринку збуту</i>	За даними Google Maps в місті Київ було виявлено 52 компанії які займаються вивезення сміття
<i>Динаміка розвитку ринку</i>	Збільшується відповідно до розростання кількості мешканців мегаполісів
<i>Існуючі обмеження для старту роботи</i>	Фінансування необхідне для розробки, впровадження, тестування та маркетингово просування проекту
<i>Специфічні вимоги та умови для отримання сертифікації</i>	Специфічні вимоги відсутні
<i>Середньостатистичні показники рентабельності в галузі</i>	Конфіденційна інформація яка не розголошується компаніями конкурентами

Існуючі обмеження для старту роботи. Основним виявленим обмеження є те, що для повноцінного запуску подібної системи необхідне значне інвестування в розробку а згодом в маркетингове просування продукту, рішенням даної проблеми може стати поступове нарощення лише необхідного функціоналу, але подібна стратегія розвитку вимагає більшої відповідальності та кращого планування з точки зору технічної архітектури системи.

Динаміка розвитку ринку. Позитивним фактором є те, що завдяки процесам глобалізації кількість населення великих міст збільшується, що в свою чергу збільшує кількість сміття, а відповідно й кількість пунктів його збору та кількість компаній які цим займаються, що для цього стартапу означає пряме збільшення кількості потенційних клієнтів.

Середньостатистичні показники рентабельності в галузі. Механізмом отримання прибутку для компанії конкурента є оформлення платної підписки з різними тарифами в залежності від об'ємів використання, та сама модель може бути використана в цьому стартапі.

Наведемо опис цільових груп проекту у таблиці 4.4.

Таблиця 4.4 – Характеристика потенційних клієнтів стартап-проекту

Потреба	Цільова аудиторія	Відмінності у поведінці цільових груп	Вимоги споживачів до товару / послуги
<i>Зменшення розриву в часі виконання між різними бригадами</i>	Водії транспортної компанії	Не виявлені	Детермінований та максимально однаковий обсяг роботи для кожної з бригад
<i>Мінімізація часових витрат під час перевезень</i>	Транспортна компанія	Не виявлені	Гарантоване покращення часу обробки зміни хоча б на 20%
<i>Зниження ціни на вивезення сміття</i>	Мешканці будинків	Не виявлені	Помітне зниження ціни на послугу вивезення сміття

Оцінивши вимоги цільових груп споживачів ми можемо визначити фактори загроз. Для того, щоб мінімізувати збитки розглянемо можливі джерела загроз та роботу, яку можна виконати для їх усунення у таблиці 4.5.

Таблиця 4.5 – Фактори загроз

Фактор	Зміст загрози	Реакція компанії
<i>Конкуренція</i>	На ринку вже присутні зрілі компанії-конкуренти які мають клієнтську базу	Запропонувати алгоритм який має кращі показники часу виконання
<i>Не зацікавленість клієнтів</i>	Більшість компаній які займаються вивезення сміття здебільшого вже використовують певні логістичні рішення, тому вони можуть бути не зацікавлені в тому, щоб впроваджувати нову інтеграцію з моїм стартап-проектом	запропонувати вигідніші тарифи ніж в конкурентів
<i>Створення конкурентами кращого алгоритму</i>	Однією з ключових особливостей проекту є кращий алгоритм складання розкладу перевезень ніж у конкурентів, тому у випадку якщо вони покращать свої алгоритми, це може стати фатальним фактором для стартапу	Постійне вкладання грошей в покращення алгоритму а також наукову діяльність
<i>Брак фінансування</i>	Недостатнє фінансування, та брак інвестицій компанії	Пошук додаткових інвестицій, а також максимальна оптимізація та мінімізація витрат.
<i>Неліквідність</i>	Для функціонування алгоритму потрібна значна обчислювальна потужність	Використовувати хмарні сервіси в яких оплата відбувається виключно в тому об'ємі в якому послуга була виконана

Також виокремимо фактори можливостей, вплив яких може позитивно сказатися на стані компаній. Дані розглянемо у таблиці 4.6.

Таблиця 4.6 – Фактори можливостей

Фактор	Зміст можливості	Реакція компанії
<i>Розширення цільової аудиторії</i>	Проект може бути адаптований для використання не лише компаніями які займаються перевезеннями сміття, а також будь-якими іншими які мають в основі своєї роботи логістичну складову	Розробка більш абстрактного функціоналу яким зможуть користуватися компанії й інших галузей спрямування
<i>Можливість встановлення монополії</i>	Завдяки найкращим тарифам та найвищій якості послуг можливість захопити лідуючу позицію на ринку та поглинути конкурентів	Розвиток бренду та захоплення максимально великої долі ринку
<i>Вихід на міжнародні ринки збуту</i>	Встановлення контрактів з зарубіжними замовниками та розширення географії ринків збуту	Локалізація продукту, інтернаціоналізація та впровадження маркетингової стратегії на ринки інших країн
<i>Економічний</i>	Підтримка інновацій у виробництві в зв'язку з чим можливе зниження вартості функціонування системи	Використання нових технологій які дозволять зменшити витрати

Проаналізувавши ринкове середовище, було виявлено п'ять факторів загроз серед яких найбільш небезпечними є не зацікавленість клієнтів та брак фінансування. Для вирішення першого з них було запропоновано створення кращої тарифної системи ніж у конкурентів, а також розробити механізми для спрощення етапу інтеграції для компаній клієнтів. Для вирішення другого фактору ризику необхідно активно проводити пошук інвестицій та завойовувати ринки суміжних галузей які задіяні в перевезенні.

Визначимо загальні риси конкуренції на ринку, для цього проаналізуємо пропозиції на ринку в таблиці 4.7.

Таблиця 4.7 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
<i>Монополія</i>	Судом доведено зловживання монопольним становищем на ринку. Відділення антимонопольного комітету дійшло висновку про економічну необґрунтованість існуючої оплати послуг.	Надання конкурентоспроможних послуг, встановлення економічно обґрунтованої ціни на послуги.
<i>Локальний рівень конкурентної боротьби</i>	Конкуренція фірм на регіональному рівні	Встановлення зв'язків із замовниками
<i>Внутрішньогалузева ознака</i>	В галузі домінують декілька конкуруючих фірм	Економічно обґрунтована ціна на послуги.
<i>Товарно-видова конкуренція</i>	Конкуренція між товарами одного виду.	Вдосконалення якості надання послуг.
<i>Цінові переваги</i>	Споживачі орієнтуються на послуги з нижчою ціною.	Технології дозволять використовувати ресурси більш раціонально, що знизить рівень витрат. Комплексна автоматизація процесу сприятиме зменшенню частки витрат на оплату праці.

Зі ступеневого аналізу конкуренції на ринку можемо зробити висновок про можливість виходу на ринок. З огляду на умови, які сформувались на ринку, розглянемо стратегію на встановлення зв'язків з наявними замовниками функцій.

Проаналізуємо умови конкуренції у галузі у таблиці 4.8.

Таблиця 4.8 – Аналіз конкуренції у галузі

Ск ла до ві ан алі зу	Конкуренти		Постачальники	Клієнти	Товари замінники
	Прямі	Потенційні			
	Ant- Logistisc	Невідомі			
			Для даного стартапу постачальниками можна вважати провайдерів хмарних технологій	Клієнти обирають логістичні рішення, які є найкращим компромісом між точністю та вартістю	Існує велика конкуренція
Ви сн ов ки	Наявні	Даних не знайдено	Умови не диктуються постачальниками послуг так, як серед них самих існує велика конкуренція	Зосередити всі зусилля для задля підвищення якості наданих послуг, на перших етапах можливе функціонування в мінус	Необхідно збільшувати кількість унікального функціоналу який відсутній у конкурентів

З огляду на проведений аналіз тип конкуренції – монополістична. Це зумовлено рядом факторів, серед яких можна виділити невжиття заходів для визначення виконавця послуг по вивозу сміття на конкурсних засадах, що призводить до недопущення конкуренції на ринку, а також майже повною відсутністю конкурентів представлених на ринку на даний момент.

Наведемо можливі фактори конкурентоспроможності у таблиці 4.9

Таблиця 4.9 – Обґрунтування факторів конкурентоспроможності

Фактор конкурентоспроможності	Обґрунтування
<i>Ціна</i>	Існуючі рішення є досить дорогими особливо для задач великої розмірності, саме тому за умови якщо моє рішення буде пропонувати кращу цінову політику ніж у конкурентів, це буде ключовою перевагою
<i>Додатковий функціонал</i>	Функція трекінгу місцезнаходження автомобілів, які у конкурентів пропонуються лише як інтеграції зі сторонніми системами
<i>Кращі часові показники</i>	Двадцяти відсоткове покращення часових показників ніж у конкурентів

Після того, як було виділено можливі фактори конкурентоспроможності, можна виокремити сильні та слабкі сторони стартап-проекту. Дані наведено у таблиці 4.10.

Таблиця 4.10 – Порівняльний аналіз сильних та слабких сторін проекту

Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів конкурентів у порівнянні з КСК						
		-3	-2	-1	0	+1	+2	+3
<i>Ціна</i>	20							+
<i>Додатковий функціонал</i>	17						+	
<i>Кращі часові показники</i>	15					+		

Підіб'ємо підсумки в ринковому аналізі можливостей впровадження проекту в матриці аналізу сильних та слабких сторін підприємства, а також його можливостей та загроз у матриці SWOT-аналізу, наведеній у таблиці 4.11.

Таблиця 4.11 – Матриця SWOT- аналізу стартап-проекту

Сильні сторони	Слабкі сторони
<ul style="list-style-type: none"> • <i>Складання маршрутів одразу для багатьох автомобілів</i> • <i>Оптимізація за часом опрацювання</i> • <i>Можливість інтеграції для клієнтів</i> • <i>Трекінг даних стосовно місцезнаходження автомобілів</i> 	<ul style="list-style-type: none"> • <i>Фінансування необхідне для розробки, впровадження, тестування та маркетингово просування проекту</i>
Можливості	Загрози
<ul style="list-style-type: none"> • <i>Розширення цільової аудиторії</i> • <i>Можливість встановлення монополії</i> • <i>Вихід на міжнародні ринки збуту</i> • <i>Економічний</i> 	<ul style="list-style-type: none"> • <i>Конкуренція</i> • <i>Не зацікавленість клієнтів</i> • <i>Створення конкурентами кращого алгоритму</i> • <i>Брак фінансування</i> • <i>Неліквідність</i>

На основі проведеного SWOT-аналізу робимо висновки про конкурентоспроможність проекту. Тепер потрібно розробити поведінку виведення стартап-проекту на ринок, спочатку проаналізуємо ймовірність отримання ресурсів за певні строки, дані наведено у таблиці 4.12.

Таблиця 4.12 – Альтернативи ринкового впровадження стартап-проекту

Альтернатива	Ймовірність отримання ресурсів	Строки реалізації
<i>Розробка, тестування та запуск системи на ринок Києва</i>	Реальна	6-8 місяців

Оскільки жодних альтернатив окрім безпосереднього впровадження системи на ринок не було виявлено, то необхідно дотримуватися цього шляху. Почати діяльність запропоновано з міста Київ.

4.5 Розроблення ринкової стратегії проекту

Тепер розробимо ринкову стратегію проекту. Виокремимо групи потенційних споживачів та опишемо їх у таблиці 4.13. На основі цих спостережень визначимо стратегію охоплення ринку.

Таблиця 4.13 – Вибір цільових груп потенційних споживачів

Опис профілю цільової групи	Готовність споживачів сприйняти послугу	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
<i>Державні/ комунальні підприємства</i>	+	Невизначений	Низька	Середня
<i>Приватні підприємства</i>	+	Середній	Висока	Складна

Цільовими групами обрано як державні так і приватні підприємства, адже обидва цих типи підприємств представлені в галузі транспортних перевезень з вивезення сміття. Для двох цільових груп ми можемо розробити диференційовану стратегію охоплення ринку, наприклад для державних підприємств можна

запропонувати спеціальні тарифи які будуть нижчими в обмін на те, що держава буде гарантувати, що даний стартап буде єдиним провайдером даних послуг з яким вона співпрацює в даній сфері.

На основі обраного ринкового сегменту сформуємо базову стратегію розвитку, дані наведено у таблиці 4.14.

Таблиця 4.14 – Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Стратегія спеціалізації	Активний маркетинг, заключення контрактів з клієнтами на більш вигідних умовах ніж у конкурентів	Кращі ціни, покращені часові характеристики для розкладу перевезень, ухил на довготривалі стосунки	Стратегія диференціації

Основною стратегією оберемо стратегію диференціації, яка орієнтується на потреби користувача. Альтернативною до неї (у разі провалу, або не досягненні очікуваного рівня фінансових показників) буде обрано стратегію спеціалізації – підлаштовування під окремий цільовий сегмент.

У таблиці 4.15 наведено обрану стратегію конкурентної поведінки на ринку.

Таблиця 4.15 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідц ем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Ні	Компанія буде шукати нових клієнтів та перехоплювати існуючих, а також захоплювати клієнтів із суміжних галузей	Компанія буде надавати той самий спектр послуг, що і конкуренти з точки зору логістика, прото буде його доповнювати такими можливостями як трекінг даних про авто в реальному часі а також пропонувати кращі характеристики для складання розкладу перевезень	Заняття конкурентної ніші

Оскільки головною діяльністю компанії є орієнтованість на клієнтів, а саме постійна підтримка та розширення функціоналу згідно до потреб клієнтів, то обраною є стратегія заняття конкурентної ніші з орієнтацією на декілька сегментів.

Означимо стратегію позиціювання продукту (таблиця 4.16).

Таблиця 4.16 – Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап- проект	Вибір асоціацій, які мають сформулювати комплексну позицію власного проекту
1. Трекінг координат транспортних засобів 2. Покращені часові показники для складеного розкладу 3. Краща цінова політика 4. Надання статистичних звітів	диференціація	Краща цінова політика та показники оптимізації за часом	1. Точність; 2. Адаптивність алгоритму 3. Клієнто-орієнтованість 4. Вигідність

Підведемо підсумки, щодо обраних стратегій:

- **обрані сегменти ринку** – державні і приватні підприємства;
- **конкурентна поведінка** – заняття конкурентної ніші;
- **альтернативна стратегія розвитку** – спеціалізація;
- **базова стратегія розвитку** – диференціація.

4.6 Розроблення маркетингової програми стартап-проекту

Після того як споживач обере зручний для нього тарифний план та проведе оплату по ньому, він отримає повний доступ до системи та зможе почати планувати розклади перевезень на основі даних про транспортні засоби та пункти збору які він ввів в систему.

Почнемо з формування маркетингової концепції товару, розрахованого на кінцевого споживача. Визначимо ключові переваги концепцій потенційного товару враховуючи попередні аналізи конкурентоспроможності, дані наведено у таблиці 4.17.

Таблиця 4.17 – Визначення ключових переваг концепцій потенційного товару

Потреба	Вигода, яку пропонує товар (послуга)	Ключові переваги перед конкурентами
Покращення часових показників сформованого розкладу перевезень	Покращений розклад та вигідніші тарифи	Наявність функції трекінгу поточного місцезнаходження транспортних засобів для будь-якого тарифного плану
Зменшення вартості надаваної послуги		

З даних, наведених у таблиці 4.17 бачимо, що товар задовольняє потреби споживачів та має переваги над конкурентами.

Для подальшого аналізу потрібно розробити трирівневу маркетингову модель товару (таблиця 4.18). В ній вкажемо ідею послуги та особливості її надання.

Таблиця 4.18 – Опис трьох рівнів моделі товару

Рівні товарів	Сутність та складові		
I. Товар за задумом	Покращений алгоритм складання розкладу перевезень для транспортних компаній		
II. Товар у Реальному виконанні	Властивості/характеристики	М/ Нм	Вр/Тх /Тл/Е/Ор
	– Якість	– Нм	– Тл
	– Точність	– Нм	– Тл
	– Відсоток покращення часових показників	– М	– Тх
	– Ціна	– Н	– Е
	– Транспортування	– М	– Тх
	Якість: в цикл розробки буде впроваджено етап обов’язкового тестування системи за сценаріями використання, а також окреме тестування для алгоритмічної частини		
	Пакування: необхідність в пакуванні відсутня		
	Марка: Для системи буде створено підприємство з однойменною назвою		
III. Товар із підкріпленням	Перший місяць безлімітного використання системи – безкоштовно для всіх зареєстрованих компаній-клієнтів		

З огляду на дані про виробництво та товари аналоги сформуємо цінові межі для потенційного товару.

Дані, щодо тарифних планів конкурентів знаходяться у відкритому доступі, завдяки чому ми можемо їх проаналізувати та зробити висновки, щодо власних тарифів які дозволять нам успішно конкурувати та будуть цікавими для нових потенційних клієнтів.

Зведемо ці дані у вигляді таблиці 4.19.

Таблиця 4.19 – Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
<p>Всі ціни вказані за місячну підписку:</p> <ul style="list-style-type: none"> – безкоштовно – до 2 маршрутів в день до 20 точок в маршруті; – 49\$ – до 5 маршрутів в день до 50 точок в маршруті; – 149\$ – до 15 маршрутів в день, необмежена кількість точок в маршруті; – 300\$ – до 50 маршрутів в день, необмежена кількість точок в маршруті. <p>Максимальний тарифний план:</p> <ul style="list-style-type: none"> – 1500\$ – до 150 маршрутів в день, необмежена кількість точок в маршруті. 	Товари аналоги не виявлені	Високий	До певного обсягу точок система буде повністю безкоштовна, максимальна вартість підписки буде складати 1000\$

Очевидним ходом для виходу на ринок при наявності сильного конкурента який на даний момент є лідером ринку – є встановлення цін, нижчим за ціни конкурентів.

Також перевагою над конкурентами може стати тариф який буде повністю безкоштовним до певного обсягу обробки даних.

На основі попереднього аналізу сформуємо систему збуту (таблиця 4.20).

Таблиця 4.20 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
<i>Система оформлення місячної підписки з диференційованою тарифною сіткою</i>	<ul style="list-style-type: none"> – розробка; – вдосконалення; – тестування; – підтримка; – відладка; – збут. 	Канал першого рівня	Пряма система збуту з пошуком клієнтів

Базуючись на стратегії позиціонування та специфіці поведінки клієнта сформуємо концепцію маркетингової комунікації (таблиця 4.21).

Таблиця 4.21 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікації	Ключові позиції для позиціонування	Концепція рекламного звернення
<i>Бажання мінімізувати витрати</i>	Цифрові ресурси; Спеціалізовані заходи (наприклад форуми урбаністичної тематики);	Канал першого рівня	Краща цінова політика, покращений алгоритм складання розкладу перевезень, забезпечення більш комфортних умов праці для водіїв транспортних компаній

В цьому підрозділі ми визначили ключові переваги та недоліки в порівнянні з конкурентами, проаналізували канали збуту, виявили основні канали комунікації з клієнтами а також, приблизно, сформували тарифні плани які будуть доступні при виході системи на ринок.

Висновок до розділу

В даному розділі було проведено формування підприємницької ідеї проекту, зроблено це було у форматі стартапу. В межах даного розділу проведено аналіз ринку на наявність конкурентів. Було виявлено головного конкурента, який має схожий функціонал. Проте в системи які пропонує даний стартап є вагомі переваги які дозволяють розраховувати на зайняття значної долі ринку.

Також було оцінено можливість технічної реалізації ідеї, було розглянуто окремі складові та сформовано вимоги необхідні для технічної реалізації для кожної з них. Проведений аналіз показав, що жодних технічних обмежень для реалізації та впровадження продукту не існує.

Оскільки проект відноситься до категорії – стартапу, а також виходить на ринок в цільовій галузі в якій вже представлені сильні конкуренти, то його можна класифікувати як досить ризикований. Для мінімізації ризиків було проведено їх детальний аналіз та проведене пошук способів їх вирішення або мінімізації.

В розділі було проаналізовано можливості для маркетингової складової, та як наслідок побудовано план маркетингово просування, що дозволить вивести стартап на ринок та вижити в умовах складної конкуренції.

ВИСНОВКИ

Під час виконання магістерської дисертації було детально розглянуто питання планування розкладу перевезень для транспортних компаній які займаються вивезенням сміття.

Було детально проаналізовано бізнес-процеси вже існуючих компаній, виявлено їх слабкі місця та розроблено пропозиції, щодо їх покращення завдяки впровадженню даної системи. Були виділені головні актори системи та встановлено вза'ємо зв'язок між ними. Також було описано постановку задачі під час якої виділено головні цілі та шляхи їх досягнення.

Відповідно до зібраних даних була сформована математична модель, а також розроблено алгоритм який вирішує задачу складання розкладів для транспортних компаній які займаються вивезенням сміття в умовах недовизначених показників, який має оптимізацію за рівномірністю розподілу роботи між усіма виконавцями. Оскільки алгоритм комплексний і є комбінацією з: алгоритму кластеризації, алгоритму побудови шляху, та алгоритму розподілу роботи, то в межах дисертації було реалізовано кожен з них. Для оцінки оптимальності алгоритму було проведено експериментальне порівняння з іншими підходами до вирішення поставленої задачі.

Систему виконано за допомогою клієнт-серверного підходу. Серверна частина виконана на мові Java з використанням фреймворку Spring, клієнтська частина виконана на мові JavaScript з використанням фреймворку VueJS. Як базу даних було обрано PostgreSQL на основі якого було спроектовано реляційну модель для системи.

Дисертація була розглянута в розрізі стартап проекту в межах якого проаналізовані ключові фактори, які необхідні врахувати при впровадженні системи та її просуванні. Систему було порівняно з потенційними конкурентами і аналіз показав, що система має набір ключових переваг які даються їй можливість успішно кокурувати.

Матеріали роботи були опубліковані на в міжвідомчому науково-технічному збірнику «Адаптивні системи автоматичного управління» №2/33 та у тезах міжнародного наукового симпозіуму «ІНТЕЛЕКТУАЛЬНІ РІШЕННЯ».

ПЕРЕЛІК ПОСИЛАНЬ

1. How does online tracking actually work? [Електронний ресурс] – Режим доступу до ресурсу: <https://robertheaton.com/2017/11/20/how-does-online-tracking-actually-work/>.
2. Google Maps Platform [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/maps/documentation>.
3. Google Maps [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Google_Maps.
4. What is OpenStreetMap? [Електронний ресурс] – Режим доступу до ресурсу: <https://derickrethans.nl/what-is-openstreetmap.html>.
5. Visicom Maps [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Visicom_Maps.
6. Yandex Search [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Yandex_Search.
7. Benish GPS [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Benish_GPS.
8. Wialon Review [Електронний ресурс] – Режим доступу до ресурсу: <https://reviews.financesonline.com/p/wialon/>.
9. What is PostgreSQL? Introduction, History, Features, Advantages [Електронний ресурс] – Режим доступу до ресурсу: <https://www.guru99.com/introduction-postgresql.html>.
10. PostgreSQL: The World's Most Advanced Open Source Relational Database [Електронний ресурс] – Режим доступу до ресурсу: <https://www.postgresql.org/about/>.
11. What is PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу: <http://www.postgresqltutorial.com/what-is-postgresql/>.
12. Нариньяні А. С. Недовизначеність в системах представлення та обробки знань //Изв. АН СССР. Техн.кибернетика. – 1986. – №5.

13. Fortin J. Gradual Numbers and their Application to Fuzzy Interval Analysis / J. Fortin, D. Dubois, H. Fargier // IEEE Transactions on Fuzzy Systems. – 2008. – 16, 2. – С. 388 – 402.
14. Эндрю А. Штучний інтелект / А. Эндрю, Д. Поспелова // Мир, 1985. – 205 с.
15. Хаггард Г. Дискретна математика для програмістів. Навчальний посібник / Г. Хаггард, Д. Шліпф, С. Уайтсайдс; Пер. з англ. Н.А. Шихова; Під ред. А.А. Сапоженков. // БИНОМ. ЛЗ, 2012. – 627 с.
16. Travelling salesman problem [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Travelling_salesman_problem.
17. Левитин А. Метод грубой силы: Задача коммивояжера. Алгоритмы: введение в разработку и анализ / А. В. Левитин // Москва: «Вильямс», 2006. – с. 159–160.
18. Гамільтонів цикл [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Гамільтонів_граф.
19. Иванов Б.Н. Дискретна математика. Алгоритми і програми. Розширений курс / Б.Н. Иванов. // М. : Известия, 2011. – 512 с.
20. NP-complete problem [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/np-completeness-set-1/>.
21. NP-completeness [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/NP-completeness>.
22. NP-complete problem [Електронний ресурс] – Режим доступу до ресурсу: <https://www.britannica.com/science/NP-complete-problem>.
23. Мандель И. Д. Кластерный анализ / И. Д. Мандель. – М.: Финансы и статистика. – 1988. – 176 с.
24. Івахненко А.Г. Объективная кластеризация на основе теории самоорганизации моделей / А.Г. Івахненко // Автоматика. – 1987. – №5. – С. 6 – 15.
25. Базилевич Р.П. Ієрархічна кластеризація – ефективний засіб розв’язування неполіноміальних комбінаторних задач схемного типу високої розмірності / Р.П. Базилевич, І.В. Подольський // "Штучний інтелект". – 2002. – №3. – С. 474– 483.
26. Simulated annealing [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Simulated_annealing.

27. Kirkpatrick S. Optimization by Simulated Annealing // S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi / Science, 1983. – P. 671-680.
28. Simulated annealing [Електронний ресурс] – Режим доступу до ресурсу: <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/learn-43/lib/photoz/.g/web/glossary/anneal.html>
29. L. M. De Campos Learning Bayesian Networks by Ant Colony Optimisation: Searching in Two Different Spaces / De Campos L. M., Gamez J. A., Puerta J. M // Mathware & Soft Computing. — 2002. — № 9.
30. Raspinelli J. M. Data Mining with an Ant Colony Optimization Algorithm / J. M. Raspinelli, H. S. Lopes, A. A. Freitas // IEEE Trans. on Evolutionary Computation. Special issue on Ant Colony Algorithms. — 2002. — V. 6. — № 4. — P. 321—332.
31. Stutzle T. MAX-MIN Ant System / T. Stutzle, H. Hoos // Future Generation Computer Systems. — 2000. — № 8(16). — С. 889—914.
32. Самойленко М. Інформаційні технології в розв'язанні транспортних задач: монографія. / М. І. Самойленко, А. О. Кобець. // Харків, Харківська національна академія міського господарства, 2011. – 256 с.
33. Давідіч Ю. Розробка графіка руху транспортних засобів при організації вантажних перевезень // Харків, ХНАМГ, 2010. – 345 с.
34. Матвейчук Т. Моделювання процесу планування вантажоперевезень / Т. Матвейчук, В. Ставицький // Вісник Національного університету "Львівська політехніка". – 2012. – С. 142-146.
35. Козаченко, Д. М. Основи дослідження операцій у транспортних системах: приклади та задачі : навч. посіб. для ВНЗ / Д. М. Козаченко, Р. В. Вернигора, В. В. Малашкін // Дніпропетровськ : ДНУЗТ, 2015. — 277 с.
36. Литвин В. В. Методика вирішення завдань пошуку оптимальних туристичних маршрутів алгоритмами наслідування мурашиної колонії / В. В. Литвин, Д. І. Угрин // Вісник Національного технічного університету "ХПІ". Серія : Інформатика та моделювання. - 2016. - № 21. - С. 47-60.
37. Алексеев С.В. Обробка інформації в складних організаційних системах // С.В. Алексеев, В.О. Мартовицький / Системи обробки інформації, 2014. – Випуск 2 (118). – С. 84-88.

38. Byte code Vs Machine code [Електронний ресурс] – Режим доступу до ресурсу:
<http://www.allaboutcomputing.net/2014/07/byte-code-vs-machine-code.html>.
39. What is the JVM? Introducing the Java Virtual Machine [Електронний ресурс] – Режим доступу до ресурсу: <https://www.javaworld.com/article/3272244/what-is-the-jvm-introducing-the-java-virtual-machine.html>.
40. JVM (Java Virtual Machine) Architecture [Електронний ресурс] – Режим доступу до ресурсу: <https://www.javatpoint.com/internal-details-of-jvm>.
41. Java Virtual Machine [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Java_virtual_machine.
42. Процеси і потоки. Багатозадачність і багатопоточність. Проблеми розробки додатків для багатопотокового середовища. [Електронний ресурс] – Режим доступу до ресурсу: <http://um.co.ua/8/8-2/8-228807.html>.
43. Путівник по Linux [Електронний ресурс] – Режим доступу до ресурсу: <https://linuxguide.rozh2sch.org.ua/>.
44. Windows [Електронний ресурс] – Режим доступу до ресурсу: <https://techterms.com/definition/windows>.
45. Що таке Docker і технологія контейнерів Linux [Електронний ресурс] – Режим доступу до ресурсу: <https://vps.ua/blog/docker-and-linux-containers/>.
- a. What is a Container? [Електронний ресурс] – Режим доступу до ресурсу: <https://aws.amazon.com/containers/>.
46. Hard Disk Drive Definition [Електронний ресурс] – Режим доступу до ресурсу: <https://www.downsizinggovernment.org/transportation/urban-transit>.
47. The ultimate explanation of the solid-state drive [Електронний ресурс] – Режим доступу до ресурсу: <https://www.digitaltrends.com/computing/what-is-an-ssd/>.
48. What is the JRE? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.javaworld.com/article/3304858/what-is-the-jre-introduction-to-the-java-runtime-environment.html>.
49. Introduction to Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.dev/>.
50. HTTP [Електронний ресурс] – Режим доступу до ресурсу: <https://www.webopedia.com/TERM/H/HTTP.html>.

- 51.What is HTTPS? [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.cloudflare.com/learning/ssl/what-is-https/>.
- 52.JSON [Електронний ресурс] – Режим доступу до ресурсу: <http://www.json.org/>.
- 53.What is Cron Job? [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.hivelocity.net/kb/what-is-cron-job/>.
- 54.What is VueJS? [Електронний ресурс] – Режим доступу до ресурсу:
<https://vuejs.org/v2/guide/>.
- 55.Everything curl [Електронний ресурс] – Режим доступу до ресурсу:
<https://ec.haxx.se/>.
- 56.Опукла оболонка [Електронний ресурс] – Режим доступу до ресурсу:
<http://techtrend.com.ua/index.php?newsid=9675> .
- 57.Обчислювальна потужність комп'ютера [Електронний ресурс] – Режим доступу до ресурсу:
https://uk.wikipedia.org/wiki/Обчислювальна_потужність_комп%27ютера.
- 58.What is CRM? [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.salesforce.com/crm/what-is-crm/>.
- 59.What is REST [Електронний ресурс] – Режим доступу до ресурсу:
<https://restfulapi.net/>.
- 60.Representational State Transfer [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Representational_state_transfer.
- 61.Polling [Електронний ресурс] – Режим доступу до ресурсу:
[https://en.wikipedia.org/wiki/Polling_\(computer_science\)](https://en.wikipedia.org/wiki/Polling_(computer_science)).
- 62.Polling [Електронний ресурс] – Режим доступу до ресурсу:
<https://enacademic.com/dic.nsf/enwiki/2208811>.
- 63.Spring Boot [Електронний ресурс] – Режим доступу до ресурсу:
<https://spring.io/projects/spring-boot>.
- 64.Introduction to Spring Boot [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.geeksforgeeks.org/introduction-to-spring-boot/>.
- 65.What is PDF? [Електронний ресурс] – Режим доступу до ресурсу:
<https://edu.gcfglobal.org/en/basic-computer-skills/what-is-a-pdf-file/1/>.

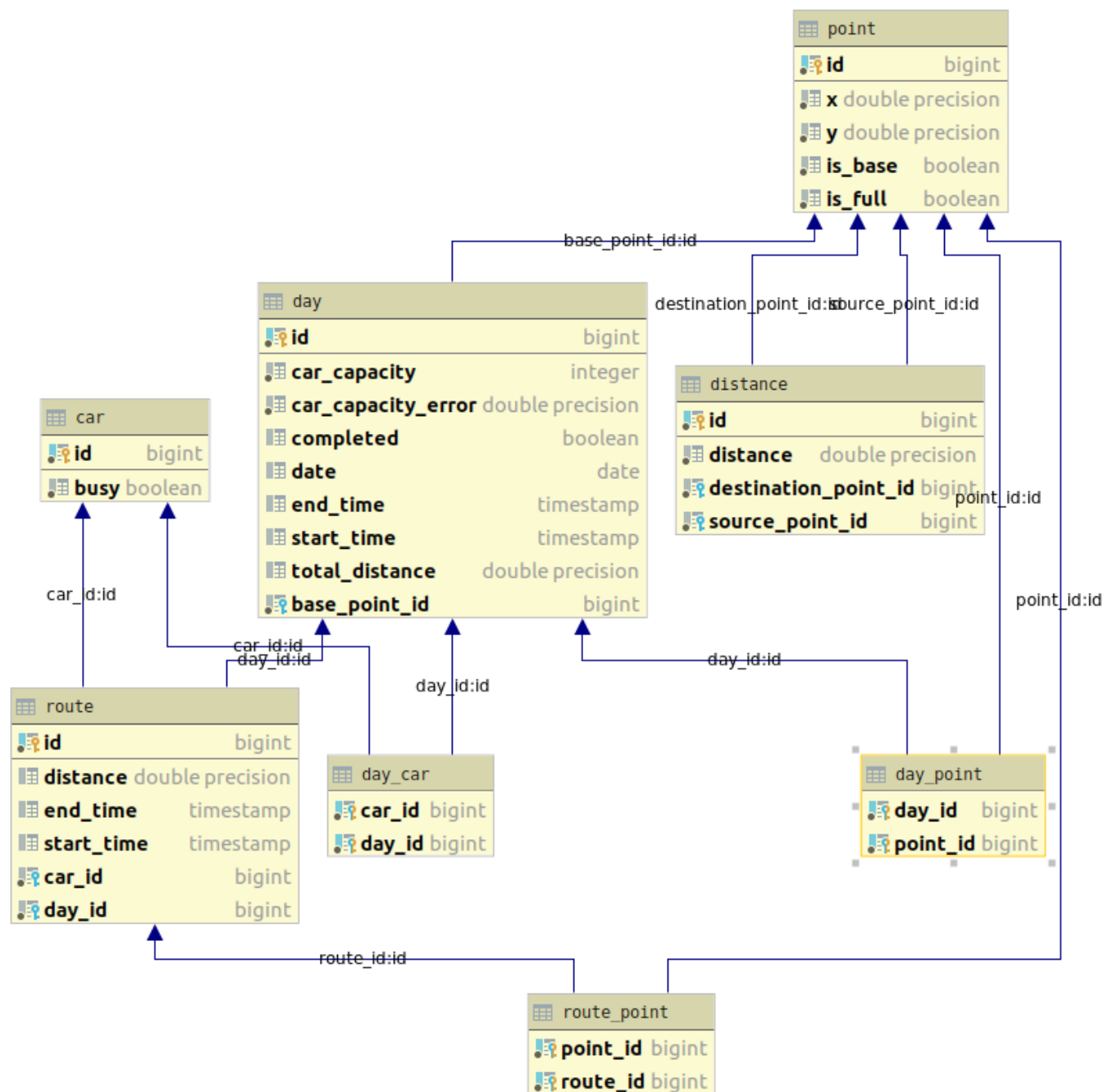
66.XLS [Электронный ресурс] – Режим доступа до ресурсу:
<https://whatis.techtarget.com/fileformat/XLS-Worksheet-file-Microsoft-Excel>.

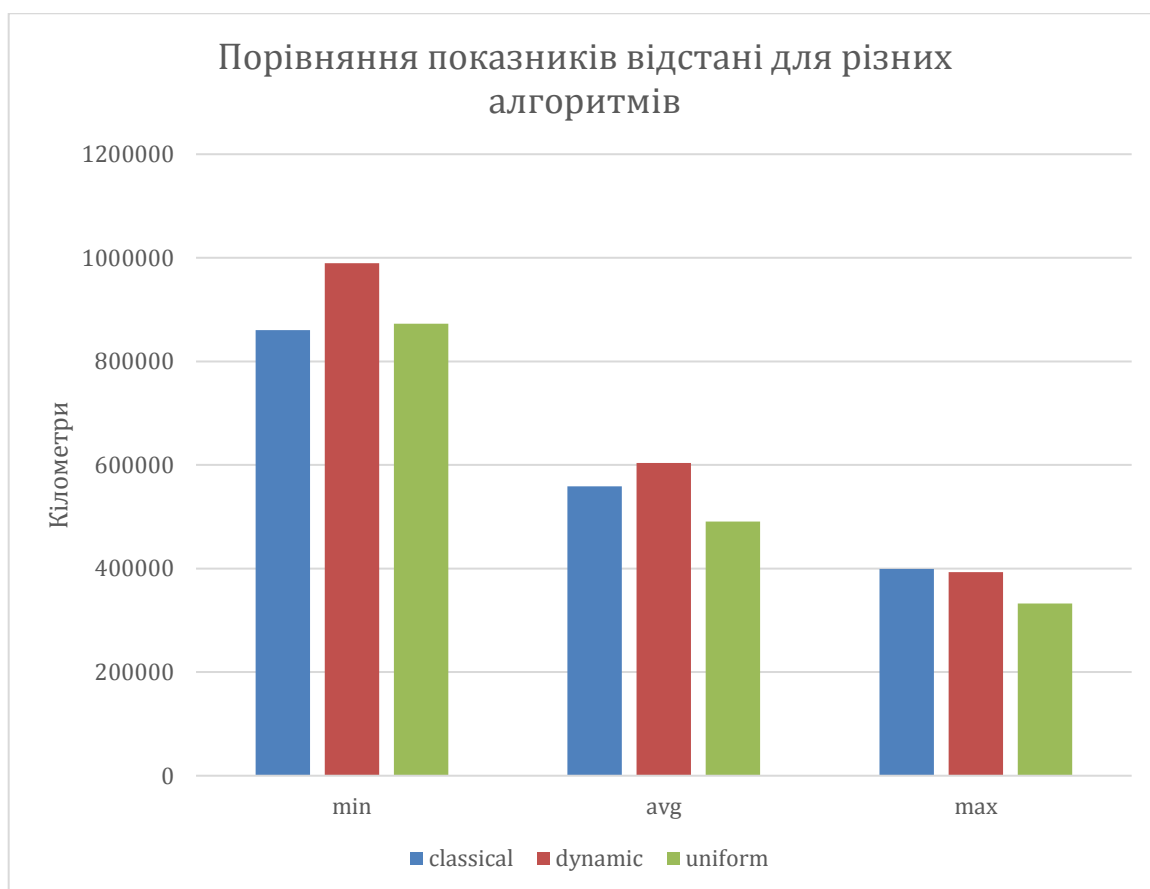
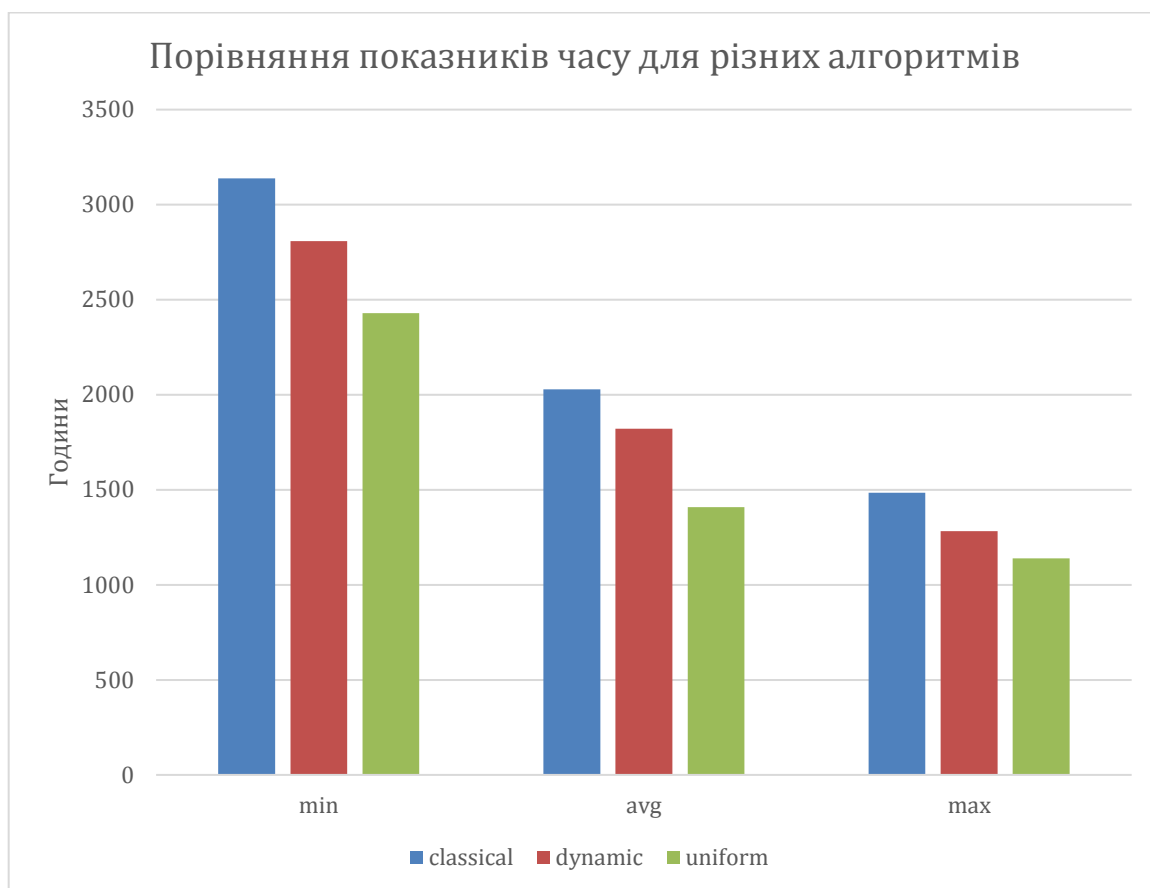
ДОДАТКИ

ДОДАТОК А Графічний матеріал

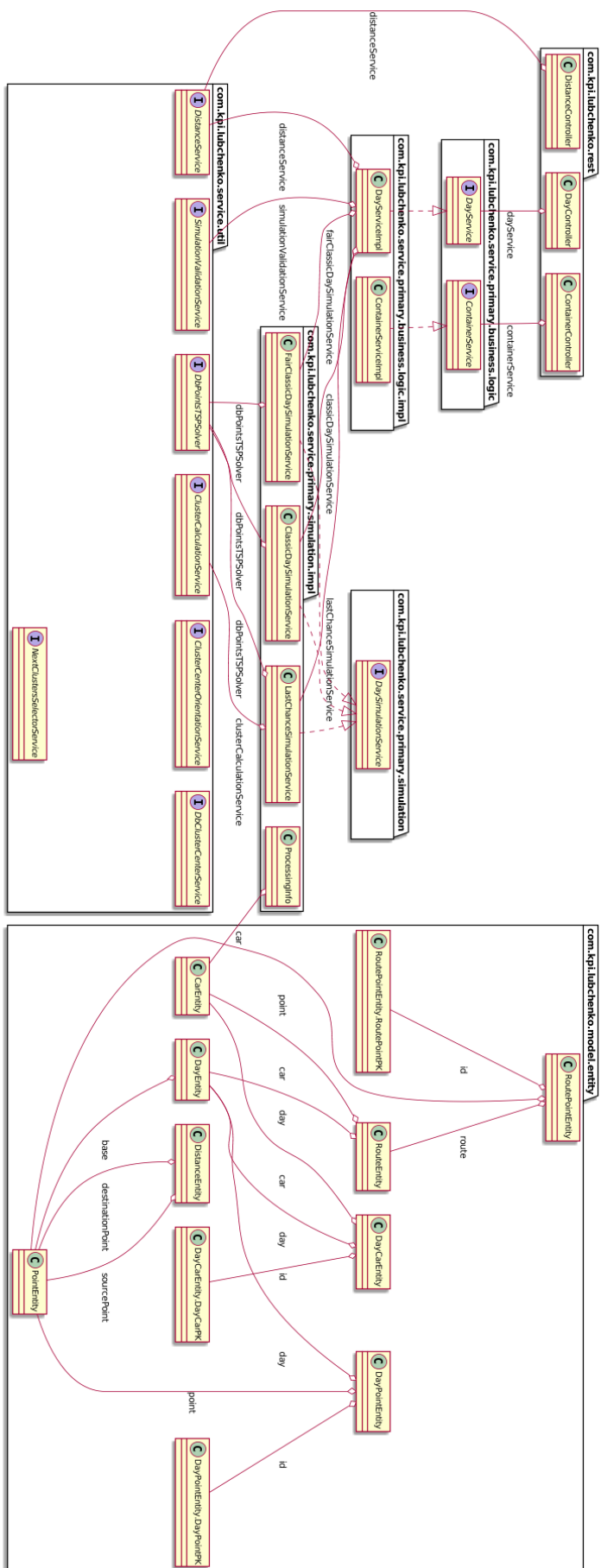


ПЛАКАТ А.2 ER-ДІАГРАМА БАЗИ ДАНИХ



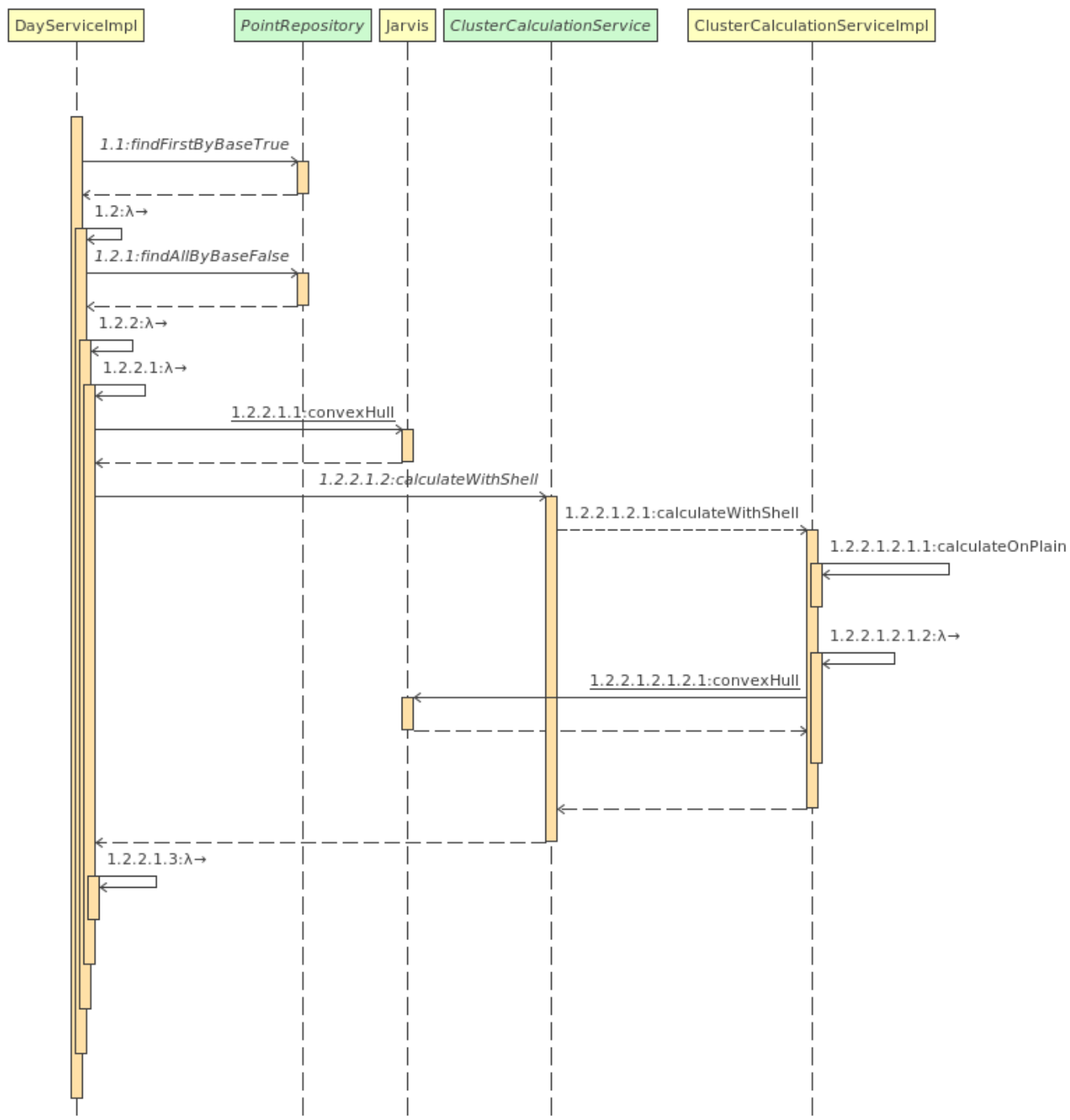
ПЛАКАТ А.3 РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТІВ

ПЛАКАТ А.4 ДІАГРАМА ПАКЕТІВ

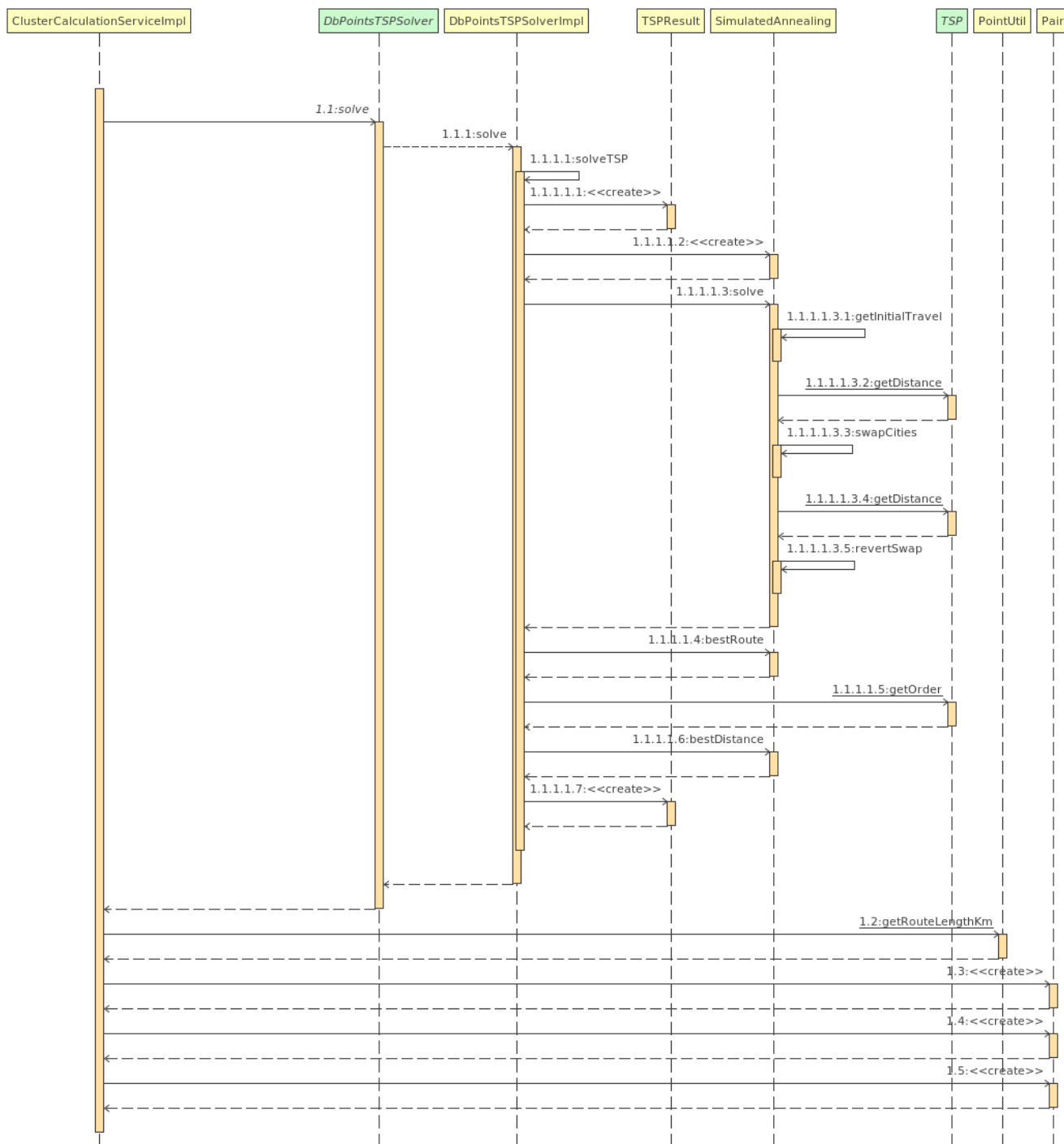


MASTERS-DISSERTATION's Class Diagram

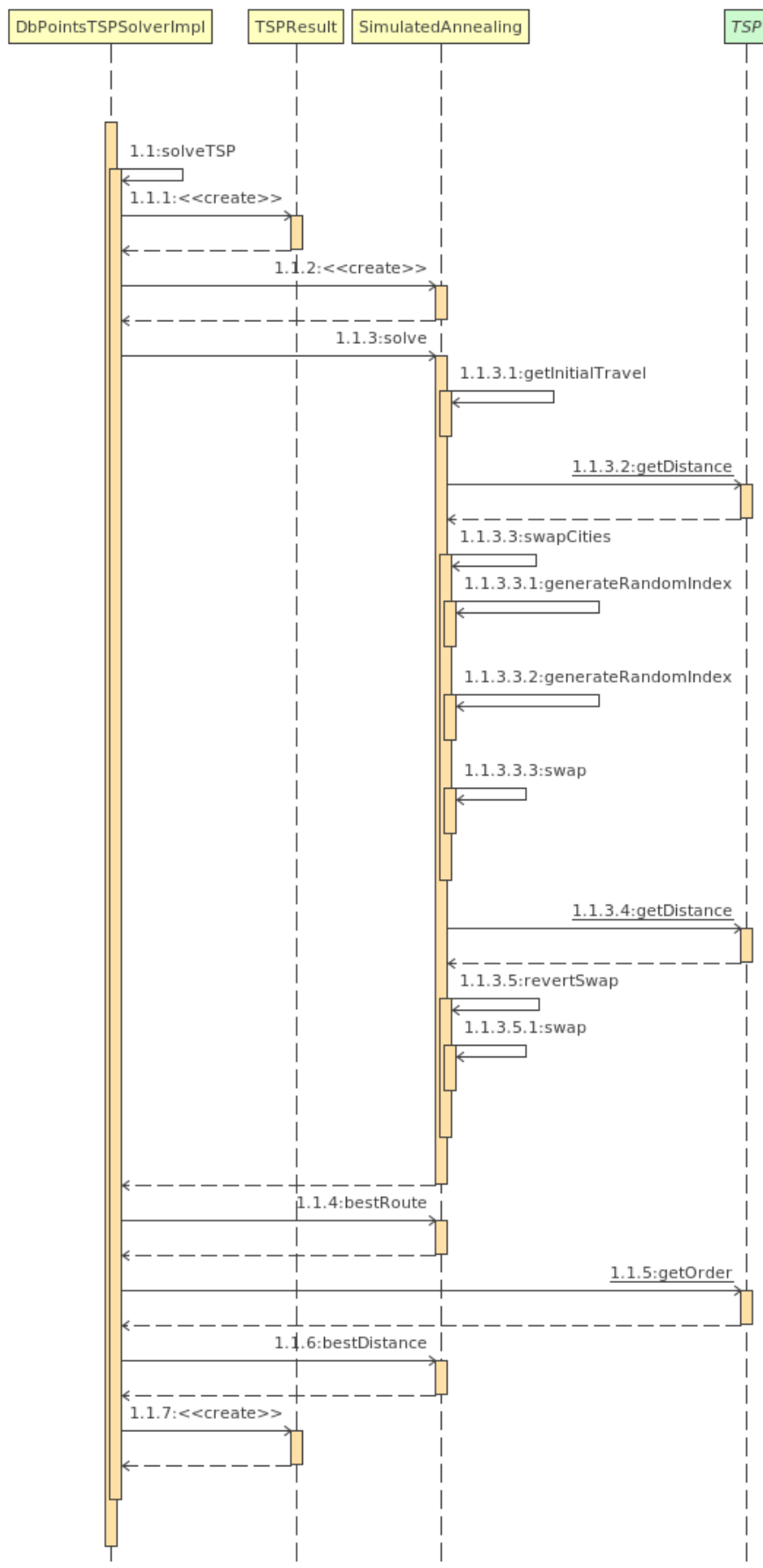
ПЛАКАТ А.5 ДІАГРАМА ПОСЛІДОВНОСТЕЙ АЛГОРИТМУ СКЛАДАННЯ РОЗКЛАДУ ПЕРЕВЕЗЕНЬ



ПЛАКАТ А.6 ДІАГРАМА ПОСЛІДОВНОСТЕЙ АЛГОРИТМУ КЛАСТЕРИЗАЦІЇ



ПЛАКАТ А.7 ДІАГРАМА ПОСЛІДОВНОСТЕЙ АЛГОРИТМУ РОЗВ'ЯЗАННЯ ЗАДАЧІ КОМІВОЯЖЕРА



ПЛАКАТ А.8 АРХИТЕКТУРА СИСТЕМИ

